Microaggregation- and Permutation-Based Anonymization of Mobility Data

Josep Domingo-Ferrer and Rolando Trujillo-Rasua

Universitat Rovira i Virgili
Department of Computer Engineering and Mathematics
UNESCO Chair in Data Privacy
Av. Països Catalans 26
E-43007 Tarragona, Catalonia
{josep.domingo,rolando.trujillo}@urv.cat

Abstract

Movement data, that is, trajectories of mobile objects, are automatically collected in huge quantities by technologies such as GPS, GSM or RFID, among others. Publishing and exploiting such data is essential to improve transportation, to understand the dynamics of the economy in a region, etc. However, there are obvious threats to the privacy of individuals if their trajectories are published in a way which allows re-identification of the individual behind a trajectory. We contribute to the literature on privacy-preserving publication of trajectories by presenting a distance measure for trajectories which naturally considers both spatial and temporal aspects of trajectories, is computable in polynomial time, and can cluster trajectories not defined over the same time span. Our distance measure can be naturally instantiated using other existing similarity measures for trajectories that are appropriate for anonymization purposes. Then, we propose two heuristics for trajectory anonymization which yield anonymized trajectories formed by fully accurate true original locations. The first heuristic is based on trajectory microaggregation using the above distance and on location permutation; it effectively achieves trajectory k-anonymity. The second heuristic is based only on location permutation; it gives up trajectory k-anonymity and aims at location k-diversity. The strong point of the second heuristic is that it takes into account reachability constraints when computing anonymized trajectories. Experimental results on a synthetic data set and a real-life data set are presented; for similar privacy protection levels and most reasonable parameter choices, our two methods offer better utility than comparable previous proposals in the literature.

Keywords: Movement data; Trajectories; Data privacy; Anonymization; Microaggregation; Permutation.

[1] The authors are with the UNESCO Chair in Data Privacy, but they are solely re-

1 Introduction

Various technologies such as GPS, RFID, GSM, etc., can sense and track the whereabouts of objects (cars, parcels, people, etc.). On the other hand, the current storage capacities allow collecting such object movement data in huge spatio-temporal databases. Analyzing this kind of databases containing the trajectories of objects can lead to useful and previously unknown knowledge. Therefore, it is beneficial to share and publish such databases and let the analysts derive useful knowledge from them —knowledge that can be applied, for example, to intelligent transportation, traffic monitoring, urban and road planning, supply chain management, sightseeing improvement, etc.

However, the privacy of individuals may be affected by the publication or the outsourcing of databases of trajectories. Several kinds of privacy threats exist. Simple de-identification realized by removing identifying attributes is insufficient to protect the privacy of individuals. The biggest threat with trajectories is the "sensitive location disclosure". In this scenario, knowing the times at which an individual visited a few locations can help an adversary to identify the individual's trajectory in the published database, and therefore learn the individual's other locations at other times. Privacy preservation in this context means that no sensitive location ought to be linkable to an individual.

The risk of sensitive location disclosure is also affected by how much the adversary knows. The adversary may have access to auxiliary information [27], also sometimes called side knowledge, background knowledge or external knowledge. The adversary can link such background knowledge obtained from other sources to information in the published database. Estimating the amount and extent of auxiliary information available to the adversary is a challenging task.

There are quite a few differences between spatio-temporal data and microdata, *i.e.*, records describing individuals in a standard database with no movement data. One real difference becomes apparent when considering privacy. Unfortunately, the traditional anonymization and sanitization methods for microdata [18] cannot be directly applied to spatio-temporal data without considerable expense in computation time and information loss. Hence, there is a need for specific anonymization methods to thwart privacy attacks and therefore

sponsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organization. This work was partly supported by the European Commission under FP7 projects "DwB" and "Inter-Trust", by the Spanish Government under projects TSI2007-65406-C03-01 "E-AEGIS", TIN2011-27076-C03-01 "CO-PRIVACY" and CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", and by the Government of Catalonia under grant 2009 SGR 01135. The first author is partly supported as an ICREA Acadèmia Researcher by the Government of Catalonia.

reduce privacy risks associated with publishing trajectories.

Trajectories can be modeled and represented in many ways [17]. Without loss of generality, we consider a trajectory to be a timestamped path in a plane. By assuming movements on the surface of the Earth, the altitude of each location visited by a trajectory stays implicit; it could be explicitly restored if the need arose. More formally, let *timestamped location* be a triple (t, x, y) with t being a timestamp and (x, y) a *location* in $\mathbb{R}^2$. Intuitively, the timestamped location denotes that at time t an object is at location (x, y).

Definition 1 (Trajectory) *A* trajectory *is an ordered set of timestamped locations*

$$T = \{(t_1, x_1, y_1), \ldots, (t_n, x_n, y_n)\} \ , \tag{1}$$

where $t_i < t_{i+1}$ *for all* $1 \leq i < n$.

Definition 2 (Sub-trajectory) *A trajectory* $S = \{(t'_1, x'_1, y'_1), \ldots, (t'_m, x'_m, y'_m)\}$ *is a* sub-trajectory *of* T *in Expression* (1)*, denoted* $S \preceq T$*, if there exist integers* $1 \leq i_1 < \ldots < i_m \leq n$ *such that* $(t'_j, x'_j, y'_j) = (t_{i_j}, x_{i_j}, y_{i_j})$ *for all* $1 \leq j \leq m$.

Hereinafter, we will use *triple* as a synonym for timestamped location. When there is no risk of ambiguity, we also say just “location” to denote a timestamped location.

1.1 Contribution and plan of this article

We present two heuristic methods for preserving the privacy of individuals when releasing trajectories. Both of them exactly preserve original locations in the sense that the anonymized trajectories contain no fake, perturbed or generalized trajectories. The first heuristic is based on microaggregation [11] of trajectories and permutation of locations. Microaggregation has been successfully used in microdata anonymization to achieve k-anonymity [39,41,13]. We use it here for trajectory k-anonymity (whereby an adversary cannot decide which of k anonymized trajectories corresponds to an original trajectory which she partly knows), first by grouping the trajectories into clusters of size at least k based on their similarity and then transforming via location permutation the trajectories inside each cluster to preserve privacy. The second heuristic aims no longer at trajectory k-anonymity, but at location k-diversity (whereby knowing a sub-trajectory S of a certain original trajectory T allows an adversary to discover a location in $T \setminus S$ with probability no greater than $1/k$); this second heuristic is based on location permutation and its strong point is that it takes reachability constraints into account: movement between locations must follow the edges of an underlying graph (*e.g.*, urban pattern) so that not all locations are reachable from any given location. Experimental results show that achieving trajectory k-anonymity with reachability constraints may not be possible without discarding a substantial fraction of locations, typically those which are rather isolated. This is the motivation for our second heuristic: it still considers reachability but it reduces the number of discarded locations by replacing k-anonymity at the trajectory level by k-diversity at the location level.

For clustering purposes, we propose a new distance for trajectories which naturally considers both spatial and temporal coordinates. Our distance is able to compare trajectories that are not defined over the same time span, without resorting to time generalization. Our distance function can compare trajectories that are timewise overlapping only partially or not at all. It may seem at first sight that the distance computation is exponential in terms of all considered trajectories, but we show that it is in fact computable in polynomial time.

We present empirical results for the two proposed heuristics using synthetic data and also real-life data. We theoretically and experimentally compare our first heuristic with a recent trajectory anonymization method called (k, δ)-anonymity [1] also aimed at trajectory k-anonymity without reachability constraints. Theoretical results show that the privacy preservation of our first method is the same as that of (k, δ)-anonymity but dealing with trajectories *not* having the same time span. For the second heuristic involving reachability constraints, no comparable counterparts seem to exist in the literature.

In summary, our contributions are:

- A distance measure for trajectories which naturally considers both spatial and temporal aspects of trajectories, is computable in polynomial time, and can cluster trajectories not defined over the same time span;
- Two methods for trajectory anonymization which yield anonymized trajectories formed by fully accurate true original locations and whose distinctive features are:
    - The first method aims at trajectory k-anonymity.
    - The second method takes reachability constraints into account, and it tries to reduce the fraction of discarded locations by replacing trajectory k-anonymity with location k-diversity;
- Empirical results on synthetic and real-life data portraying the performance of the two above methods. Both methods are confronted with (k, δ)-anonymity [1], which has some comparable features.

This paper extends the workshop paper [12], which presented an anonymization method aimed at forming anonymized trajectories with true original locations and providing high utility properties but without a proven privacy level. Here, we leverage the idea presented in [12] of trajectory anonymization by means of location permutation, and we propose two new methods that effectively achieve proven privacy levels. Furthermore, in [12] empirical results were obtained only on synthetic data, while in this paper we extend empirical results by using a real-life data set of trajectories.

The rest of this article is organized as follows. Section 2 reviews related work. Section 3 describes the utility features, the adversarial model being considered and the target privacy properties. Our new distance between trajectories is described in Section 4. Our two new anonymization methods are specified in Section 5. Their privacy guarantees are examined in Section 6. Section 7 reports on empirical results. Conclusions are drawn in Section 8.

2 Related work

Most trajectory anonymization methods in the literature rest on ideas inspired by microdata anonymization. We first recall two key microdata anonymization concepts: k-anonymity and microaggregation. We then review the trajectory anonymization literature. We end this section by reviewing similarity distance measures and clustering algorithms for trajectories.

2.1 k-Anonymity and microaggregation

A lot of work has been done in anonymizing microdata and relational/transactional databases [39,41,45,30,46,28,35,15,48]; see also the recent survey [18]. A usual goal in anonymization is to achieve k-anonymity [39,41], which is the "safety in numbers" notion.

Anonymizing a microdata set by mere suppression of direct identifiers (*e.g.*, names, passport numbers) is not enough to prevent privacy disclosure. Indeed, other attributes, called quasi-identifier attributes, are often available in the data set such that their combination allows re-identifying the individual to whom a record corresponds: for example, Sweeney [42] found that "87% of the US population is uniquely identified by {date of birth, gender, 5-digit ZIP}". Re-identification allows linking the confidential attributes in a record (*e.g.*, salary or health condition) with a specific individual, and this constitutes a disclosure.

An anonymized microdata set is said to satisfy k-anonymity if each combination of quasi-identifier attribute values is shared by at least k records. Therefore, this property guarantees that an adversary is unable to identify the individual to whom an anonymized record corresponds with probability higher than $1/k$.

k-Anonymity cannot be directly achieved with spatio-temporal data, because any point or time can be regarded as a quasi-identifier attribute [1]. Direct k-anonymization would require a set of original trajectories to be transformed into a set of anonymized trajectories such that each of the latter is identical to at least $k-1$ other anonymized trajectories. This would obviously cause a huge information loss.

Generalization was the computational approach originally proposed to achieve k-anonymity [39,41]. Later, Zhang *et al.* introduced the permutation-based approach [48], that has the advantage of not being constrained by domain generalization hierarchies. In [13] it was shown that k-anonymity could also be achieved through microaggregation of quasi-identifiers. Microaggregation [11] works in two stages:

(1) *Clustering.* The original records are partitioned into clusters based on some similarity measure (some kind of distance) among the records with the restriction that each cluster must contain at least k records. Several microaggregation heuristics are available in the literature, some yielding fixed-size clusters all of size k, except perhaps one (*e.g.* the MDAV heuristic [13]), and some yielding variable-size clusters, of sizes between k and $2k-1$ (*e.g.* μ-Approx [14]). We will use fixed-size microaggregation.

(2) *Anonymization.* Each cluster is anonymized individually. Anonymization of a cluster may be based on an aggregation operator like the average [11] or the median [13], which is used to compute the cluster centroid; each record in the cluster is then replaced by the cluster centroid. Anonymization of a cluster can also be achieved by replacing the records in the cluster with synthetic or partially synthetic data; this is called hybrid data microaggregation [10] or condensation [3].

To use microaggregation on trajectories, we need a distance measure to compute the similarity between trajectories. We deal with possible distances later in this paper.

2.2 Trajectory anonymization

Just like in microdata records, suppressing direct identifiers from trajectories is not enough for privacy [26]. Consequently, several anonymity notions and methods for trajectories have been proposed [21,20,22,7,37,6,1,33,43,31,34,47,32,2,23,24]. Among those works, we next review the ones that are most similar to our approach, and we highlight our comparative advantages. Other comparisons of several trajectory anonymization methods can be found in [6,2].

Closest to our approach is the notion of (k, δ)-anonymity [1,2]. In the original method –Never Walk Alone (NWA) [1]–, the set of trajectories is partitioned into disjoint subsets in which trajectories begin and end at roughly the same time; then trajectories within each set are clustered using the Euclidean distance. In the follow-up method –Wait For Me (W4M) [2]–, the original trajectories are clustered using the edit distance on real sequences (EDR) [9]. Both approaches proceed by anonymizing each cluster separately. Two trajectories T_1 and T_2 are said to be co-localized with respect to δ in a certain time interval $[t_1, t_n]$ if for each triple (t, x_1, y_1) in T_1 and each triple (t, x_2, y_2) in T_2 with $t \in [t_1, t_n]$, it holds that the spatial Euclidean distance between both triples is not greater than δ. Anonymity in this context means that each trajectory is co-localized with at least $k-1$ other trajectories. Anonymization is achieved by spatial translation of trajectories inside a cluster of at least k trajectories having the same time span. In the special case when $\delta = 0$, the method produces one centroid/average trajectory that represents each and all trajectories in the cluster. *Ad hoc* preprocessing and outlier removal facilitate the process. Utility is evaluated in terms of trajectory distortion and impact on the results of range queries. The problem with the NWA method is that partitioning the set of all trajectories into subsets sharing the same time span may produce too many subsets with too few trajectories inside each of them; clearly, a subset with less than k trajectories cannot be k-anonymized. Also, setting a value for δ may be awkward in many applications, *e.g.* trajectories recorded using RFID technology. In Section 7 we present an empirical comparison between this method and our two heuristics. Our heuristics avoid the above subset problem by considering all trajectories together whatever their time span; they also achieve co-localization without requiring a δ radius. The W4M method is similar in clustering to our clustering, although it uses the EDR distance between trajectories, which has the shortcomings discussed further below.

Another k-anonymity based notion for trajectories consisting of ranges of points and ranges of times has been proposed in [33] and [34]. It uses clustering to minimize the "log cost metric"; this balances the spatial and temporal translations with user-provided weights. Minimizing the log cost therefore maximizes utility. The clusters are anonymized by matching points of the trajectories and generalizing them into minimum bounding boxes. Unmatched points are suppressed and so are some trajectories. The anonymized data are not released; instead, synthetic "atomic" trajectories (having unit x-range, y-range and time range) are generated by sampling the bounding boxes. This approach does not release standard trajectories but only trajectories with unit ranges. In comparison, we are able to produce synthetic trajectories, with the advantage that we obtain anonymized trajectories formed by true original locations.

In [32], k-anonymity means that an original trajectory T is generalized into a trajectory $g(T)$ (without the time information) in such a way that $g(T)$ is a sub-trajectory of the generalizations of at least $k-1$ other original trajectories. Ignoring the time information during anonymization and complex plane tessellations used to achieve the k-anonymity are the main drawbacks of this method. Utility is measured by comparing clustering results. In our approach, we avoid complex tessellations and our main advantage in comparison to this anonymization scheme is that we do not ignore temporal information.

Another proposal for achieving k-anonymity of trajectories by means of generalization is [24]. The difference lies in the way generalization is performed: the authors propose a technique called local enlargement, guaranteeing that user locations are enlarged just enough to reach k-anonymity, which improves utility of the anonymized trajectories. In contrast, we preserve original locations, without generalizing them; our notion of trajectory k-anonymity is, however, reformulated as discussed below.

The adapted k-anonymity notion for trajectories in [47] is stated in terms of a bipartite attack graph relating original and anonymized trajectories such that the graph is symmetric and the degree of each vertex representing an anonymized trajectory is at least k. The quasi-identifiers used to define identities are the times of the positions in a trajectory, and the anonymity is achieved by generalizing points of trajectories into areas on the grid. An information loss metric defined for such areas is used to evaluate the utility of the anonymized data.

Some approaches assume that the data owner anonymizing the database knows exactly what the adversary's knowledge is. If the adversary is assumed to know different parts of trajectories, then those are removed from the published data [43]. However, this work only considers sequential place visitation without real timestamps. If the adversary is assumed to use some prediction of continuation of a trajectory based on previous path and speed, then uncertainty-aware path cloaking [22,23] can suppress these trajectories; this however results in high information loss.

In contrast to these methods, we perform traditional microaggregation over all original trajectories —we do not specially and separately consider trajectories having the same time span and we consider trajectories over locations, not ranges, without stripping the time information. We publish synthetic trajectories which are analogous to condensed or hybrid microdata [3,10]. However, our synthetic trajectories are formed by locations covered by the original trajectories. This means that the location points of our anonymized trajectories remain on the underlying network map.

Additional related work about anonymization of spatio-temporal data can be found in the literature about location privacy, focused on applications such as privacy-aware location-based services (LBS) or privacy-aware monitoring of continuously moving objects. Location privacy in the LBS-setting was first proposed in [19]. See [36,25] for recent papers on location privacy, in which mobile objects protect the privacy of their continuous movement. Location privacy is enforced on individual sensitive locations or unlinked locations in an on-line mode; often, data are anonymized on a per-request basis and in the context of obtaining a location-based service. In this article, we focus on off-line publishing whole spatio-temporal databases rather than protecting specific individuals from LBS providers or on-line movement monitoring. In general, a solution to location privacy is not a solution for publishing anonymized trajectories, and vice versa.

2.3 Trajectory similarity measures

As argued in Section 2.1 above, using microaggregation for trajectory k-anonymization requires a distance function to measure the similarity between trajectories. Such a distance function must consider both space and time. Although most spatial distances can be extended into spatio-temporal distances by adding a time co-ordinate to spatial points, it is not obvious how to balance the weight of spatial and temporal dimensions. Furthermore, not all similarity measures for trajectories are suitable for comparing trajectories for anonymization purposes. The requirement for anonymization is not just similarity regarding shape, but also spatial and temporal closeness. Some typical distances for trajectories include the Euclidean distance, the Hausdorff distance [40], the Fréchet distance [4], the turning point distance [5], and distances based on time series [29] —*e.g.*, dynamic time warping (DTW), short time series (STS)— and on edit distance [9] —*e.g*, edit distance with real penalty (ERP), longest common sub-sequence (LCSS), and the edit distance on real sequences (EDR) discussed next.

The *edit distance on real sequences* (EDR) [9] is the number of insert, delete, or replace operations that are needed to change one sequence into another. If P and Q are two sequences of m and n triples, respectively, where each triple λ has three attributes – x-position $\lambda.x$, y-position $\lambda.y$ and time $\lambda.t$ – the distance $EDR(P,Q)$ is defined as:

$$\begin{cases} \max\{m,n\} & \text{if } m=0 \text{ or } n=0 \\ \min\{match(p_1,q_1)+EDR(Rest(P),Rest(Q)), & \text{otherwise} \\ \quad 1+EDR(Rest(P),Q), 1+EDR(P,Rest(Q))\} & \end{cases}$$

where p_1 and q_1 are the first elements of a given sequence, $Rest(\cdot)$ is a function that returns the input sequence without the first element, and where $match(p,q) := 0$ if p and q are "close", that is, they satisfy either $|p.x -$

$q.x| \leq \epsilon$ and $|p.y - q.y| \leq \epsilon$ for some parameter ϵ [9] or $|p.x - q.x| \leq \Delta.x$, $|p.y - q.y| \leq \Delta.y$, and $|p.t - q.t| \leq \Delta.t$ for a triple of parameters Δ [2]; otherwise, $match(p, q) := 1$. This definition of *match* means that the cost for one insert, delete, or replace operation in EDR is 1 if p and q are not "close".

EDR has been employed for anonymization in [2]. However, the edit distance and variations thereof are not suitable to guide clustering for anonymization purposes. Indeed, Figure 1 shows trajectories with different degrees of "closeness" to trajectory A, but whose EDR distance from A is the same in all cases. When timestamps are considered, the situation is even worse.

In Section 4, we define a distance measure which is better suited for anonymization clustering: it can compare trajectories defined over different time spans and even trajectories that are time-wise non-overlapping.

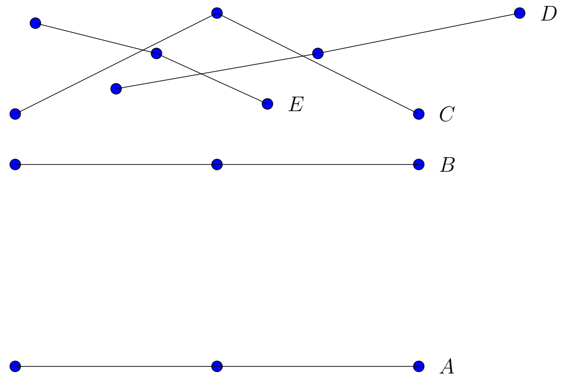


Fig. 1. Trajectories B, C, D, E are placed at varying "closeness" from A, yet their EDR distance from A is 3 in all cases. We assume that the first point of A matches the first point of each of B, C, D, E; also, second points are assumed to match each other, and the same for third points.

3 Utility and privacy requirements

Every trajectory anonymization algorithm must combine utility and privacy. However, utility and privacy are two largely antagonistic concepts. What is useful in a set of trajectories is application-dependent, so for each utility feature probably a different anonymization algorithm is needed.

3.1 Desirable utility features [1]

The utility features that are usually considered in trajectory anonymization are: (i) trajectory length preservation, (ii) trajectory shape preservation, (iii) trajectory time preservation, and (iv) minimization of the number of discarded locations. We include two additional utility features that are particularly meaningful in urban scenarios:

- *Location preservation.* This essentially means that no fake or inaccurate locations are used to replace original locations; otherwise put, locations in the anonymized trajectories should be locations visited by the original trajectories, without any generalization or accuracy loss. Preserving original locations helps answering several queries that may not be responded by generalization methods [32] or some microaggregation methods [1,2]: i) what is the ranking of original (non-removed) locations, from most visited to least visited?; ii) in which original (non-removed) locations did two or more mobile objects meet?, etc. On the other hand, if trajectory anonymization rests on replacing true locations with fake locations, an adversary can distinguish the latter from the former and discard fake locations. Hence, location preservation is desirable for both utility and privacy reasons.
- *Reachability.* In the second proposed heuristic, easy reachability between two successive locations in each anonymized trajectory is enforced. This means that the distance from the i-th location to the $i+1$-th location on an anonymized location *following the underlying network of streets and/or roads* should be at most R^s, where R^s is a preset parameter. Like location preservation, this is as good for utility as it is for privacy: if the adversary sees that reaching the $i+1$-th location from the i-th takes a long trip across streets and roads, she will guess that the section between those two locations was not present in any original trajectory.

3.2 Specific utility measures

Basic utility measures are the number of removed trajectories and the number of removed locations, whether during pre-processing, clustering or cluster anonymization.

The distortion of the trajectory shape is another utility measure, which can be captured with the space distortion metric [1, Sec.VI.B]. This metric also allows accumulating the total space distortion of all anonymized trajectories from original ones.

Definition 3 (Space distortion metric [1]) *The space distortion of an anonymized trajectory $T^\star$ with respect to its original trajectory T at time t when T has triple (t,x,y) and $T^\star$ has possible triple $(t,x^\star,y^\star)$, is*

$$SD_t(T,T^\star) = \begin{cases} \Delta((x,y),(x^\star,y^\star)) & \textit{if } (x^\star,y^\star) \textit{ is defined at } t \\ \Omega & \textit{otherwise} \end{cases}$$

*where Δ is a distance (*e.g.*, Euclidean), and Ω a constant that penalizes for removed locations. The space distortion of an anonymized trajectory $T^\star$ from its original T is then*

$$SD(T, T^\star) = \sum_{t \in TS} SD_t(T, T^\star) \ ,$$

where TS are all the timestamps where T is defined. In particular, if T is discarded during anonymization, $T^\star$ is empty, and so $SD(T, T^\star) = n\Omega$, where $n = |TS|$ is the number of locations of T. In this way, the space distortion of a set of trajectories $\mathcal{T}$ from its anonymized set $\mathcal{T}^\star$ is easily defined as

$$TotalSD(\mathcal{T}, \mathcal{T}^\star) = \sum_{T \in \mathcal{T}} SD(T, T^\star) \ ,$$

where $T^\star \in \mathcal{T}^\star$ (which may be empty) corresponds to $T \in \mathcal{T}$.

Another way to measure utility is by comparing the results between queries performed on both the original data set $\mathcal{T}$ and the anonymized data set$\mathcal{T}^\star$. Intuitively, when results on both data sets are similar for a large and diverse number of queries, the anonymized data set can be regarded as preserving the utility of the original data set. The challenge of this utility measure is the selection of queries, which is usually application-dependent or even user-dependent, *i.e.* two different users are likely to perform different queries on the same trajectory data set.

In [44] six types of spatio-temporal range queries were introduced, aimed at evaluating the relative position of a moving object with respect to a region R in a time interval $[t_b, t_e]$. We have used these queries in our experimental work, even though they were designed for use on uncertain trajectories (see Definition 4) rather than synthetic trajectories.

Definition 4 (Uncertain trajectory) *Given a trajectory T and an uncertainty space threshold σ, an* uncertain trajectory *$U(T, \sigma)$ is defined as the pair $< T, \sigma >$, where $(t, x, y) \in U(T, \sigma)$ if and only if $\exists x', y'$ such that $(t, x', y') \in T$ and the Euclidean distance between (x, y) and (x', y') is not greater than σ.*

Definition 5 (Possible motion curve) *A* possible motion curve *PMC^T of an uncertain trajectory $U(T, \sigma)$ is an ordered set of timestamped locations*

$$PMC^T = \{(t_1, x_1, y_1), \dots, (t_n, x_n, y_n)\} \ , \tag{2}$$

such that $(t_i, x_i, y_i) \in U(T, \sigma)$ for all $1 \leq i \leq n$.

In short, a possible motion curve defines one of the possible trajectories that an object moving along an uncertain trajectory could follow. Unlike in [44], our anonymized trajectories are not uncertain; hence, we will only use the two spatio-temporal range queries proposed in that paper that can be adapted to non-uncertain trajectories:

- *Sometime_Definitely_Inside(T, R, t_b, t_e)* is *true* if and only if there exists a time $t \in [t_b, t_e]$ at which every possible motion curve PMC^T of an uncertain trajectory $U(T, \sigma)$ is inside region R. For a non-uncertain T, the previous condition can be adapted as: if and only if there exists a time $t \in [t_b, t_e]$ at which T is inside R.
- *Always_Definitely_Inside(T, R, t_b, t_e)* is *true* if and only if at every time $t \in [t_b, t_e]$, every possible motion curve PMC^T of an uncertain trajectory $U(T, \sigma)$ is inside region R. For a non-uncertain T, the previous condition becomes: if and only if at every time $t \in [t_b, t_e]$, trajectory T is inside R.

3.3 Adversarial model and target privacy properties

In our adversarial model, the adversary has access to the published anonymized set of trajectories $\mathcal{T}^\star$. Furthermore, the adversary also knows that every location $\lambda \in \mathcal{T}^\star$ must be in the original set ot trajectories $\mathcal{T}$. Note that this adversary's knowledge makes an important difference from previous adversarial models [1,34,32,47], because in our model the linkage of some location with some user reveals the exact location of this user rather than a generalized or perturbed location.

Further, the method used for transforming the original set of trajectories $\mathcal{T}$ into $\mathcal{T}^\star$ is assumed known by the adversary. However, this does not include the method parameters or the seeds for pseudo-random number generators, which are considered secret. Indeed, the two methods we are proposing rely on random permutations of locations and random selection of trajectories during the clustering process, and such randomness is in practice implemented using pseudo-random number generators. If an adversary knew the seeds of the generators, she could easily reconstruct the original trajectories from the anonymized trajectories.

Finally, the adversary also knows a sub-trajectory S of some original target trajectory $T \in \mathcal{T}$ ($S \preceq T$) and knows that the anonymized version of T is in $\mathcal{T}^\star$. As in previous works, we consider that every location in $\mathcal{T}$ is sensitive, *i.e.* for any location, learning that a specific user visited it represents useful knowledge for the adversary.

Then, we identify two attacks:

(1) Find a trajectory $T^\star \in \mathcal{T}^\star$ that is the anonymized version of T.
(2) Given a location $\lambda \notin S$, determine whether $\lambda \in T$.

If the adversary succeeds in the first attack of linking a trajectory $T^\star$ with the target T, the second is not trivial, because in general the locations in $T^\star$ will not be those in T, but it is indeed easier. This means that both attacks are not independent. However, the second attack can trivially succeed even if the first attack does not: if all anonymized trajectories cross the same location λ and $\lambda \notin S$, the adversary knows that $\lambda \in T$. As we show below, both attacks are related to the two well-known privacy notions of k-anonymity [39,41] and ℓ-diversity [30], respectively.

Definition 6 (Trajectory p-privacy) *Let $Pr_{T^\star}[T|S]$ denote the probability of the adversary's correctly linking the anonymized trajectory $T^\star \in \mathcal{T}^\star$ with T given the adversary's knowledge $S \preceq T$. Then,* trajectory p-privacy *is met when $Pr_{T^\star}[T|S] \leq p$ for every trajectory $T \in \mathcal{T}$ and every subset $S \preceq T$.*

Definition 7 (Trajectory k-anonymity) *Trajectory k-anonymity is achieved if and only if trajectory $\frac{1}{k}$-privacy is met.*

Definition 8 (Location p-privacy) *Let $Pr_\lambda[T|S]$ denote the probability of the adversary's success in correctly determining a location $\lambda \in T \setminus S$, given the adversary's knowledge $S \preceq T$. Then,* location p-privacy *is met when $Pr_\lambda[T|S] \leq p$ for every triple (T, S, λ) such that $T \in \mathcal{T}$, $S \preceq T$ and $\lambda \notin S$.*

Definition 9 (Location k-diversity) *Location k-diversity is achieved if and only if location $\frac{1}{k}$-privacy is met.*

3.4 Discussion on privacy models

Achieving straightforward trajectory k-anonymity, where each anonymized trajectory would be identical to $k-1$ other anonymized trajectories, would in general cause a huge information loss. This is why some other trajectory k-anonymity definitions under different assumptions have been proposed.

The (k, δ)-anonymity definition [1,2] relies on the uncertainty inherent to trajectory data recorded by technologies like GPS. However, it may be hardly applied when accurate data sets of trajectories are needed. Furthermore, in order to achieve (k, δ)-anonymity, the k identical anonymized trajectories should be defined roughly in the same interval of time and they must contain the same number of locations. Such constraints are indeed hard to meet.

Another trajectory k-anonymity definition can be found in [37]. In this work, trajectory k-anonymity is achieved when there are at least k anonymized trajectories in $\mathcal{T}^\star$ having an anonymized version of T as a sub-trajectory. Although this definition ignores the time dimension, it does not require the length of the k anonymized trajectories to be equal. However, suppose that the adversary has a trajectory T consisting of only one location, an individual's home; whatever the anonymization method, the anonymized version of T is likely to be very similar to T. This means that there will be k anonymized trajectories containing the single location of T. However, not all these anonymized trajectories start at the single location of T. Since an individual's home is likely to be the first location of any individual's original trajectory, those anonymized trajectories that do not start at the single location of T (just pass through it) can be filtered out by an adversary and only the remaining trajectories are considered. The same filtering process can be performed if the adversary knows locations where the individual has never been. In this way, using side knowledge the adversary identifies less than k anonymized trajectories compatible with the original trajectory T. Hence, this definition may not actually guarantee k-anonymity in the sense of Definition 7.

In conclusion, different levels of privacy can be provided according to different assumptions on the original data, the anonymized data, and the adversary's capabilities. We defined above trajectory p-privacy (Definition 6) and location p-privacy (Definition 8) in order to capture two different privacy notions when the original locations are preserved.

4 Distance between trajectories

Clustering trajectories requires defining a similarity measure —a distance between two trajectories. Because trajectories are distributed over space and time, a distance that considers both spatial and temporal aspects of trajectories is needed. Many distance measures have been proposed in the past for both trajectories of moving objects and for time series but, as discussed in Section 2, most of them are ill-suited to compare trajectories for anonymization purposes. Therefore we define a new distance which can compare trajectories that are only partially or not at all timewise overlapping. We believe this is necessary to cluster trajectories for anonymization. We need some preliminary notions.

4.1 Contemporary and synchronized trajectories

Definition 10 (p%-contemporary trajectories) *Two trajectories*

$$T_i = \{(t_1^i, x_1^i, y_1^i), \ldots, (t_n^i, x_n^i, y_n^i)\}$$

and

$$T_j = \{(t_1^j, x_1^j, y_1^j), \ldots, (t_m^j, x_m^j, y_m^j)\}$$

are said to be p%-contemporary if

$$p = 100 \cdot \min(\frac{I}{t_n^i - t_1^i}, \frac{I}{t_m^j - t_1^j})$$

with $I = \max(\min(t_n^i, t_m^j) - \max(t_1^i, t_1^j), 0)$.

Intuitively, two trajectories are 100%-contemporary if and only if they start at the same time and end at the same time; two trajectories are 0%-contemporary if and only if they occur during non-overlapping time intervals. Denote the overlap time of two trajectories T_i and T_j as $ot(T_i, T_j)$.

Definition 11 (Synchronized trajectories) *Given two p%-contemporary trajectories T_i and T_j for some $p > 0$, both* trajectories are said to be synchronized *if they have the same number of locations timestamped within $ot(T_i, T_j)$ and these correspond to the same timestamps. A* set of trajectories is said to be synchronized *if all pairs of p%-contemporary trajectories in it are synchronized, where $p > 0$ may be different for each pair.*

If we assume that between two locations of a trajectory, the object is moving along a straight line between the locations at a constant speed, then interpolating new locations is straightforward. Trajectories can be then synchronized in the sense that if one trajectory has a location at time t, then other trajectories defined at that time will also have a (possibly interpolated) location at time t. This transformation guarantees that the set of new locations interpolated in order to synchronize trajectories is of minimum cardinality. Algorithm 1 describes this process. The time complexity of this algorithm is $O(|TS|^2)$ where $|TS|$ is the number of different timestamps in the data set.

Algorithm 1 Trajectory synchronization

```
Require: $\mathcal{T} = \{T_1, \ldots, T_N\}$ a set of trajectories to be synchronized, where
    each $T_i \in \mathcal{T}$ is of the form:
        $T_i = \{(t_1^i, x_1^i, y_1^i), \ldots, (t_{n^i}^i, x_{n^i}^i, y_{n^i}^i)\};$
Let $TS = \{t_j^i \mid (t_j^i, x_j^i, y_j^i) \in T_i \;:\; T_i \in \mathcal{T}\}$ be all timestamps from all
    locations of all trajectories;
for all $T_i \in \mathcal{T}$ do
    for all $ts \in TS$ with $t_1^i < ts < t_{n^i}^i$ do
        if location having timestamp $ts$ is not in $T_i$ then
            insert new location in $T_i$ having the timestamp $ts$ and coordinates
                interpolated from the two timewise-neighboring locations;
        end if
    end for
end for
```

Definition 12 (Distance between trajectories) *Consider a set of synchronized trajectories $\mathcal{T} = \{T_1, \ldots, T_N\}$ where each trajectory is written as*

$$T_i = \{(t_1^i, x_1^i, y_1^i), \ldots, (t_{n^i}^i, x_{n^i}^i, y_{n^i}^i)\} \ .$$

The distance between trajectories *is defined as follows. If $T_i, T_j \in \mathcal{T}$ are p%-contemporary with $p > 0$, then*

$$d(T_i, T_j) = \frac{1}{p}\sqrt{\sum_{t_\ell \in ot(T_i,T_j)} \frac{(x_\ell^i - x_\ell^j)^2 + (y_\ell^i - y_\ell^j)^2}{|ot(T_i, T_j)|^2}} \ .$$

If $T_i, T_j \in \mathcal{T}$ are 0%-contemporary but there is at least one subset of $\mathcal{T}$

$$\mathcal{T}^k(ij) = \{T_1^{ijk}, T_2^{ijk}, \ldots, T_{n^{ijk}}^{ijk}\} \subseteq \mathcal{T}$$

such that $T_1^{ijk} = T_i$, $T_{n^{ijk}}^{ijk} = T_j$ and T_ℓ^{ijk} and $T_{\ell+1}^{ijk}$ are p_ℓ%-contemporary with $p_\ell > 0$ for $\ell = 1$ to $n^{ijk} - 1$, then

$$d(T_i, T_j) = \min_{\mathcal{T}^k(ij)} \left(\sum_{\ell=1}^{n^{ijk}-1} d(T_\ell^{ijk}, T_{\ell+1}^{ijk}) \right)$$

Otherwise $d(T_i, T_j)$ is not defined.

The computation of the distance between every pair of trajectories is not exponential as it could seem from the definition. Polynomial-time computation of a distance graph containing the distances between all pairs of trajectories can be done as follows.

Definition 13 (Distance graph) *A* distance graph *is a weighted graph where*

(i) Nodes represent trajectories,
(ii) two nodes T_i and T_j are adjacent if the corresponding trajectories are p%-contemporary for some $p > 0$, and
(iii) the weight of the edge (T_i, T_j) is the distance between the trajectories T_i and T_j.

Now, given the distance graph for $\mathcal{T} = \{T_1, \ldots, T_N\}$, the distance $d(T_i, T_j)$ for two trajectories is easily computed as the minimum cost path between the nodes T_i and T_j, if such path exists. The inability to compute the distance for all possible trajectories (the last case of Definition 12) naturally splits the distance graph into connected components. The connected component that has the majority of the trajectories must be kept, while the remaining components represent outlier trajectories that are discarded in order to preserve privacy. Finally, given the connected component of the distance graph having the majority of the trajectories of $\mathcal{T}$, the distance $d(T_i, T_j)$ for *any two* trajectories on this connected component is easily computed as the minimum cost path between the nodes T_i and T_j. The minimum cost path between every pair of nodes can be computed using the Floyd-Warshall algorithm [16] with computational cost $O(N^3)$, *i.e.*, in polynomial time.

4.3 Intuition and rationale of the distance

In order to deal with the time dimension, our distance measure applies a linear penalty of $\frac{1}{p}$ to those trajectories that are p%-contemporary. This means that, the closer in time are two trajectories, the shorter is our distance between both. It should be remarked that we choose a linear penalty because the Euclidean distance is also linear in terms of the spatial coordinates and the Euclidean distance is the spatial distance measure we consider by default in this work. Other distances and other penalties might be chosen, *e.g.* $\frac{1}{p^2}$.

A problem appears when considering 0%-contemporary trajectories. How can two non-overlapping trajectories be penalized? A well-known strategy is to give a weight to the time dimension and another weight to the spatial dimension. By doing so, the time distance and the spatial distance can be computed separately, and later be merged using their weights. However, determining proper values for these weights is a challenging task.

Anyway, the following lemma guarantees that, whenever we consider two trajectories at minimum distance for clustering, they do have some overlap.

Lemma 1 *Any two trajectories in data set $\mathcal{T}$ at minimum distance are p%-contemporary with $p > 0$.*

Proof: Consider a trajectory $T_i \in \mathcal{T}$ and another trajectory $T_j \in \mathcal{T}$ at minimum distance from T_i. Assume that T_i and T_j are not p%-contemporary with $p > 0$. Then, since the distance between T_i and T_j is defined, according to Definition 12 a subset of distinct trajectories $\mathcal{T}(ij) = \{T_1^{ij}, T_2^{ij}, \ldots, T_{n^{ij}}^{ij}\} \subseteq \mathcal{T}$ must exist such that $T_1^{ij} = T_i$, $T_{n^{ij}}^{ij} = T_j$ and T_ℓ^{ij} and $T_{\ell+1}^{ij}$ are p_ℓ%-contemporary with $p_\ell > 0$ for $\ell = 1$ to $n^{ij} - 1$, and

$$d(T_i, T_j) = \sum_{\ell=1}^{n^{ij}-1} d(T_\ell^{ij}, T_{\ell+1}^{ij})$$

Then $d(T_i, T_j) > d(T_\ell^{ij}, T_{\ell+1}^{ij})$ for all ℓ from 1 to $n^{ij} - 1$ (strict inequality holds because all trajectories in $\mathcal{T}(ij)$ are distinct). Thus, we reach the contradiction that $d(T_i, T_j)$ is not minimum. Hence, the lemma must hold. □

5 Anonymization methods

We present two anonymization methods, called SwapLocations and ReachLocations, respectively, both of which yield anonymized trajectories formed by original locations. The first of them is partially based on microaggregation [11] of trajectories and partially based on permutation of locations. The second method is based on permutation of locations. The main difference between the SwapTriples method [12] and the two new methods we propose here is that the latter effectively guarantee trajectory k-anonymity (SwapLocations) or location k-diversity (ReachLocations). To that end, an original triple is discarded if it cannot be swapped randomly with another triple drawn from a set of $k-1$ other original triples.

Our two methods differ from each other in several aspects. The first method assumes an unconstrained environment, while the second one considers an environment with mobility constraints, like an underlying street or road network. SwapLocations effectively achieves trajectory k-anonymity. ReachLocations provides higher utility by design, but regarding privacy, it offers location k-diversity instead of trajectory k-anonymity. A common feature of both methods is that locations in the resulting anonymized trajectories are true, fully accurate original locations, *i.e.* no fake, generalized or perturbed locations are given in the anonymized data set of trajectories.

5.1 The SwapLocations method

Algorithm 2 describes the process followed by the SwapLocations method in order to anonymize a set of trajectories. First, the set of trajectories is partitioned into several clusters. Then, each cluster is anonymized using the SwapLocations function in Algorithm 3. We should remark here that we only consider trajectories for which the distance to other trajectories can be computed using the distance in Definition 12. Otherwise said, given the distance graph G (Definition 13), our distance measure can only be used within one of the connected components of G; obviously, we take the trajectories in the largest connected component of G. It should also be remarked that Algorithm 1 is only used for computing the distance between trajectories. Once a cluster C is created, the anonymization algorithm works over the original triples of the trajectories in C, and not over the triples created during synchronization.

We limit ourselves to clustering algorithms which try to minimize the sum of the intra-cluster distances or approximate the minimum and such that the cardinality of each cluster is k, with k an input parameter; if the number of trajectories is not a multiple of k, one or more clusters must absorb the up to $k-1$ remaining trajectories, hence those clusters will have cardinalities between $k+1$ and $2k-1$. This type of clustering is precisely the one used in microaggregation [11]. The purpose of minimizing the sum of the intra-cluster distances is to obtain clusters as homogeneous as possible, so that the subsequent independent treatment of clusters does not cause much information loss. The purpose of setting k as the cluster size is to fulfill trajectory k-anonymity, as shown in Section 6.1. We employ any microaggregation heuristic for clustering purposes (see Section 2 and details in Section 5.3 below).

Algorithm 2 Cluster-based trajectory anonymization($\mathcal{T}, R^t, R^s, k$)

```
Require: i) $\mathcal{T} = \{T_1, \ldots, T_N\}$ a set of original trajectories such that $d(T_i, T_j)$ is defined for all $T_i, T_j \in \mathcal{T}$, ii) $R^t$ a time threshold and $R^s$ a space threshold;
Use any clustering algorithm to cluster the trajectories of $\mathcal{T}$, while minimizing the sum of intra-cluster distances measured with the distance of Definition 12 and ensuring that minimum cluster size is $k$;
Let $C_1, C_2, \ldots, C_{n_\mathcal{T}}$ be the resulting clusters;
for all clusters $C_i$ do
    $C_i^\star$ = SwapLocations($C_i, R^t, R^s$);                // Algorithm 3
end for
Let $\mathcal{T}^\star = C_1^\star \cup \cdots \cup C_{n_\mathcal{T}}^\star$ be the set of anonymized trajectories.
```

The SwapLocations function (Algorithm 3) begins with a random trajectory T in C. The function attempts to cluster each unswapped triple λ in T with another $k-1$ unswapped triples belonging to different trajectories such that: i) the timestamps of these triples differ by no more than a time threshold R^t from the timestamp of λ; ii) the spatial coordinates differ by no more than a space threshold R^s. If no $k-1$ suitable triples can be found that can be clustered with λ, then λ is removed; otherwise, random swaps of triples are performed within the formed cluster. Randomly swapping this cluster of triples guarantees that any of these triples has the same probability of remaining in its original trajectory or becoming a new triple in any of the other $k-1$ trajectories. Note that Algorithm 3 guarantees that every triple λ of every trajectory $T \in C$ will be swapped or removed.

The SwapLocations function specified by Algorithm 3 swaps entire triples, that is, time and space coordinates. The following example illustrates the advantages of swapping time together with space.

Example 1 Imagine John attended one day the political protests in Tahrir Square, Cairo, Egypt, but he would not like his political views to become broadly known. Assume John's trajectory is anonymized and published. Assume further that an adversary knows the precise time John left his hotel in the morning, say 6:36 AM, *e.g.* because the adversary has bribed the hotel concierge into recording John's arrival and departure times. Now:

Algorithm 3 SwapLocations(C, R^t, R^s)

```
Require: i) $C$ a cluster of trajectories to be transformed, ii) $R^t$ a time threshold and $R^s$ a space threshold;
 1: Mark all triples in trajectories in $C$ as "unswapped";
 2: Let $T$ be a random trajectory in $C$;
 3: for all "unswapped" triples $\lambda = (t_\lambda, x_\lambda, y_\lambda)$ in $T$ do
 4:   Let $U = \{\lambda\}$; // Initialize $U$ with $\{\lambda\}$
 5:   for all trajectories $T'$ in $C$ with $T' \neq T$ do
 6:     Look for an "unswapped" triple $\lambda' = (t_{\lambda'}, x_{\lambda'}, y_{\lambda'})$ in $T'$ minimizing
        the intra-cluster distance in $U \cup \{\lambda'\}$ and such that:
            $|t_{\lambda'} - t_\lambda| \leq R^t$
            $0 \leq \sqrt{(x_{\lambda'} - x_\lambda)^2 + (y_{\lambda'} - y_\lambda)^2} \leq R^s$ ;
 7:     if $\lambda'$ exists then
 8:       $U \leftarrow U \cup \{\lambda'\}$;
 9:     else
10:       Remove $\lambda$ from $T$;
11:       Goto line 3 in order to analyze the next triple $\lambda$;
12:     end if
13:   end for
14:   Randomly swap all triples in $U$;
15:   Mark all triples in $U$ as "swapped";
16: end for
17: Remove all "unswapped" triples in $C$;
18: return $C$.
```

- If SwapLocations swapped only spatial coordinates, the adversary could re-identify John's trajectory as one starting with a triple (6:36 AM, x'_h, y'_h). Furthermore, (x'_h, y'_h) must be a location within a distance R^s from the hotel coordinates (x_h, y_h), although the adversary does not know the precise value of R^s. The re-identified trajectory would contain all true timestamps of John's original trajectory (because they would not have been swapped), and spatial coordinates within distance R^s of John's really visited spatial coordinates. Hence, it would be easy to check whether John was near Tahrir Square during that day. Without swapping times, privacy protection can only be obtained by taking R^s large enough so that within distance R^s of the original locations visited by John there are several semantically different spatial coordinates. To explain what we mean by semantic difference, assume (x, y) is Tahrir Square and the trajectory anonymizer guarantees that he has taken R^s large enough so that (x, y) could be swapped with some spatial coordinates (x', y') off Tahrir Square; even if (x', y') turned out to be still within Tahrir Square, John could claim to have been off Tahrir Square; the adversary could not disprove such a claim, because in fact (x, y) could be at a distance R^s from (x', y') and hence outside the Square. However, a large

R^s means a large total space distortion.

- If entire triples are swapped, as actually done by SwapLocations, the adversary can indeed locate an anonymized trajectory containing (not necessarily starting with) triple (6:36 AM, x_h, y_h). However, there is only a chance $1/k$ that this triple was not swapped from another of the $k-1$ original trajectories with which John's original trajectory was clustered. Similarly, the other triples in the anonymized trajectory containing (6:36 AM, x_h, y_h) have also most likely "landed" in that anonymized trajectory as a result of a swap with some location in some of the $k-1$ original trajectories clustered with John's. Hence, John's trajectory is cloaked with $k-1$ other trajectories. We will prove in Section 6.1 that this guarantees trajectory k-anonymity in the sense of Definition 7. In particular, the triple (t, x, y) corresponding to John at Tahrir Square will appear in one of the k anonymized trajectories, unless that triple has been removed by the SwapLocations function because it was unswappable (the smaller R^t and R^s, the more likely it is for the triple to be removed).

5.2 The ReachLocations method

The ReachLocations method, described in Algorithm 4, takes reachability constraints into account: from a given location, only those locations at a distance below a threshold *following a path in an underlying graph* (*e.g.*, urban pattern or road network) are considered to be directly reachable. Enforcing such reachability constraints while requiring full trajectory k-anonymity would result in a lot of original locations being discarded. To avoid this, trajectory k-anonymity is changed by another useful privacy definition: location k-diversity.

Computationally, this means that trajectories are *not* microaggregated into clusters of size k. Instead, each location is k-anonymized independently using the entire set of locations of all trajectories. To do so, a cluster C_λ of "unswapped" locations is created around a given location λ, *i.e.* $\lambda \in C_\lambda$. The cluster C_λ is constrained as follows: i) it must have the lowest intra-cluster distance among those clusters of k "unswapped" locations that contain the location λ; ii) it must have locations belonging to k different trajectories; and iii) it must contain only locations at a path from λ at most R^s long and with timestamps differing from t_λ at most R^t. Then, the spatial coordinates (x_λ, y_λ) are swapped with the spatial coordinates of some random location in C_λ and both locations are marked as "swapped". If no cluster C_λ can be found, the location λ is removed from the data set and will not be considered anymore in the subsequent anonymization. This process continues until no more "unswapped" locations appear in the data set.

It should be remarked that, according to Algorithm 4, two successive locations

λ_j^i and λ_{j+1}^i of an original trajectory T_i may be cloaked with respective sets of $k-1$ locations belonging to different sets of $k-1$ original trajectories; this is why we cannot speak of trajectory k-anonymity, see the example below.

Example 2 Consider $k-1$ trajectories within city A, $k-1$ trajectories within city B and one trajectory T_{AB} crossing from A to B. When applying ReachLocations, the initial locations of T_{AB} are swapped with locations of trajectories within A, whereas the final locations of T_{AB} are swapped with locations of trajectories within B. Imagine that an adversary knows a sub-trajectory S of T_{AB} containing one location λ_A in A and one location λ_B in B. Assume λ_A and λ_B are not removed by ReachLocations anonymization. Now, the adversary will know that the anonymized trajectory $T_{AB}^{\star}$ corresponding to T_{AB} is the only anonymized trajectory crossing from A to B. Thus, there is no trajectory k-anonymity, even if the adversary will be unable to determine the exact locations of $T_{AB} \setminus S$, because each of them has been swapped within a set of k locations.

Algorithm 4 swaps only spatial coordinates instead of full triples. We show in the example below that this is enough for ReachLocations to achieve location k-diversity (we have shown above that it cannot achieve trajectory k-anonymity anyway). If swapping time coordinates is not beneficial in terms of privacy guarantees, they should not be swapped, because the fact that anonymized trajectories preserve the original sequence of timestamps of original trajectories increases their utility.

Example 3 Let us resume Example 1, but now assume that ReachLocations is used instead of SwapLocations to anonymize trajectories. In this case, the adversary will find an anonymized trajectory starting with (6:36 AM,x'_h,y'_h). This anonymized trajectory will contain all true timestamps of John's original trajectory. However, the spatial coordinates appearing in any location of this re-identified trajectory are John's original spatial coordinates with a probability at most $1/k$. We will prove in Section 6.2 below that this guarantees location k-diversity in the sense of Definition 9. If we want to prevent the adversary from making sure that John visited Tahrir Square, we should take R^s large enough (the discussion in Example 1 about the protection afforded by a large R^s when time is not swapped is valid here).

5.3 Complexity of SwapLocations and ReachLocations

We first give a complexity assessment of SwapLocations and ReachLocations assuming that the distance graph mentioned in Section 4.2 has been precomputed and is available. This is reasonable, because the distance graph needs to be computed only once, while the anonymization methods may need to be

Algorithm 4 ReachLocations($\mathcal{T}, G, R^t, R^s, k$)

```
Require: i) $\mathcal{T} = \{T_1, \ldots, T_N\}$ a set of original trajectories, ii) $G$ a graph describing the paths between locations, iii) $R^t$ is a time threshold and $R^s$ is a space threshold;
Let $TL = \{\lambda_j^i \in T_i \ : \ T_i \in \mathcal{T}\}$ contain all locations from all trajectories, where $\lambda_j^i = (t_j^i, x_j^i, y_j^i)$ and the spatial coordinates $(x_j^i, y_j^i)$ are called a point;
Mark all locations in $TL$ as “unswapped”;
Let $\mathcal{T}^\star = \emptyset$ be an empty set of anonymized trajectories;
while there exist trajectories in $\mathcal{T}$ do
  Let $T_i$ be a trajectory randomly chosen in $\mathcal{T}$;
  for $j = 1$ to $j = |T_i|$ do
    if $\lambda_j^i$ is “unswapped” then
      Let $C_j^i = \{\lambda_1, \cdots, \lambda_{k-1}\}$ be a cluster of locations in $TL$ such that:
      (1) All locations in $C_j^i$ are “unswapped”, with points different from $(x_j^i, y_j^i)$ and no two equal points;
      (2) Points in $C_j^i$ belong to trajectories in $\mathcal{T} \setminus \{T_i\}$ and no two points belong to the same trajectory;
      (3) For any $\lambda \in C_j^i$, it holds that:
          (a) $|t_\lambda - t_j^i| \leq R^t$;
          (b) If $j > 1$ there is a path in $G$ between $(x_{j-1}^i, y_{j-1}^i)$ and $(x_\lambda, y_\lambda)$;
          (c) If $j < |T_i|$ there is a path in $G$ between $(x_\lambda, y_\lambda)$ and $(x_{j+1}^i, y_{j+1}^i)$;
          (d) The length of each path above is no more than $R^s$;
      (4) The sum of intra-cluster distances (following paths in $G$) in $C_j^i \cup \{\lambda_j^i\}$ is minimum among clusters of cardinality $k-1$ meeting the previous conditions;
      if such a cluster $C_j^i$ does not exist then
        Remove $\lambda_j^i$ from $T_i$;
      else
        Mark $\lambda_j^i$ as “swapped”;
        With probability $\frac{k-1}{k}$:
          (1) Pick a random location $\lambda \in C_j^i$ and mark it as “swapped”;
          (2) Swap the spatial coordinates $(x_j^i, y_j^i)$ of $\lambda_j^i$ with the spatial coordinates $(x_\lambda, y_\lambda)$ of $\lambda$;
      end if
    end if
  end for
  $\mathcal{T}^\star = \mathcal{T}^\star \cup \{T_i\}$;
  Remove $T_i$ from $\mathcal{T}$;
end while
return $\mathcal{T}^\star$.
```

run several times (*e.g.* with different parameters). Regarding SwapLocations, we have:

- Algorithm 2 can use any fixed-size microaggregation heuristic for clustering (*e.g.* MDAV in [13]). Most microaggregation heuristics have quadratic complexity, that is $O(N^2)$, where N is the number of trajectories.
- Algorithm 2 calls the procedure SwapLocations once for each resulting cluster, that is, $O(N/k)$ times.
- In the worst case, the complexity of procedure SwapLocations (Algorithm 3) is proportional to the number of locations of the longest trajectory in C, say $O(n_{max})$. For each location, a search of another location for swapping is performed among the other $k-1$ trajectories. The number of candidates for swapping is $O((k-1)n_{max})$. Hence, the complexity of SwapLocations is $O((k-1)n_{max}^2)$.
- The total complexity of the method is thus

$$O(N^2) + O(N/k) \cdot O((k-1)n_{max}^2) = O(N^2) + O(Nn_{max}^2) \qquad (3)$$

Regarding the complexity of ReachLocations, we have

- Algorithm 4 has an external loop which is called N times, where N is the number of trajectories in $\mathcal{T}$. For each trajectory, a swap is attempted for each of its unswapped locations. Hence the algorithm performs $O(Nn_{max})$ swaps, where n_{max} is the number of locations in the longest trajectory.
- Each swap involves forming a cluster which $k-1$ locations selected from TL, which takes time proportional to the total number of locations in TL, that is, $O(Nn_{max})$.
- Hence, the total complexity of the method is $O(N^2 n_{max}^2)$.

By comparing the last expression and Expression (3), we see that both SwapLocations and ReachLocations are quadratic in N and quadratic in n_{max}, but ReachLocations is slower. Such complexity motivates the following two comments related to scalability:

- If the number of trajectories N in the original data set is very large, quadratic complexity may be very time consuming. In this case, a good strategy is to use some blocking technique to split the original data set into several subsets of trajectories, each of which should be anonymized separately.
- n_{max} being large may be less problematic than N being large, provided that only a small fraction of trajectories have n_{max} or close to n_{max} locations. If a lot of trajectories are very long, a good strategy would be to split each of these into two or more trajectories and anonymize them independently.

Finally, in case we add the time complexity of the computation of the distance graph mentioned in Section 4.2 (which is $O(N^3)$ using the Floyd-Warshall algorithm), the time complexities of both SwapLocations and ReachLocations become $O(N^3) + O(Nn_{max}^2)$ and $O(N^3) + O(N^2n_{max}^2)$, respectively.

6 Privacy guarantees

6.1 Privacy guarantees of SwapLocations

The main difference between the SwapTriples method in [12] and the SwapLocations method here is that, in the latter, no original location survives unswapped in an anonymized trajectory.

Proposition 1 *Let $S \preceq T_S$ be the adversary's knowledge of a target original trajectory T_S and $\lambda_1, \lambda_2, \cdots, \lambda_{|S|}$ be all triples in S. For every trajectory T_i, the probability that the triple λ in S appears in the anonymized version $T_i^\star$ of T_i produced by SwapLocations is:*

$$\Pr(\lambda \in T_i^\star | \lambda \in S) = \begin{cases} \frac{1}{k} & \textit{if } T_S \textit{ and } T_i \textit{ lie in the same cluster} \\ \\ 0 & \textit{otherwise.} \end{cases}$$

Proof: By construction of Algorithm 3, if T_S and T_i do not lie in the same cluster, there is no possibility of swapping triples between them. Hence, in this case, $\Pr(\lambda \in T_i^\star | \lambda \in S) = 0$.

Let $T_1, T_2, \cdots, T_k \in \mathcal{T}$ be k trajectories that are anonymized together in the same cluster by the SwapLocations method. Without loss of generality, let us assume that $T_S = T_1$. By construction of Algorithm 3, for every $1 \leq i \leq k$, $\Pr(\lambda \in T_i^\star | \lambda \in T_1)$ is 0 if λ was removed, $\frac{1}{k}$ otherwise. Note that a swapping option is to swap a triple with itself, that is, not to swap it. Since it does not make sense to consider removed triples in S, we conclude that $\Pr(\lambda_j \in T_i^\star | \lambda_j \in T_1) = \frac{1}{k}, \ \forall 1 \leq j \leq |S|, 1 \leq i \leq k$ and, in consequence, $\Pr(\lambda_j \in T_i^\star | \lambda_j \in S) = \frac{1}{k}, \ \forall 1 \leq j \leq |S|, 1 \leq i \leq k$. □

Theorem 1 *The SwapLocations method achieves trajectory k-anonymity.*

Proof: By Proposition 1, any sub-trajectory $S' \preceq S \preceq T_1$ has the same probability of being a sub-trajectory of $T_1^\star$ than of being a sub-trajectory of any of the $k-1$ trajectories $T_2^\star, \cdots, T_k^\star$. Thus, given S, an adversary is not able to link T_1 with $T_1^\star$ with probability higher than $\frac{1}{k}$. Therefore, SwapLocations satisfies $\frac{1}{k}$-privacy according to Definition 6; according to Definition 7, it also satisfies trajectory k-anonymity. □

We show below that ReachLocations provides location k-diversity.

Proposition 2 *Any triple λ in an original trajectory T appears in the anonymized trajectory $T^\star$ corresponding to T obtained with ReachLocations if and only if λ was not removed and was swapped with itself, which happens with probability at most $\frac{1}{k}$.*

Proof: Let us prove the necessity implication. By construction of Algorithm 4, any triple λ whose spatial coordinates (point) cannot be swapped within a cluster $C \cup \{\lambda\}$ containing k different points belonging to k different trajectories is removed and does not appear in the set of anonymized trajectories. Further, the only way for a non-removed triple $\lambda \in T$ to survive unaltered in $T^\star$ is precisely that its point is swapped with itself, which happens with probability $\frac{1}{k}$. Therefore, to survive unaltered in $T^\star$, a triple in T needs to avoid removal and to have its point swapped with itself, which happens with probability at most $\frac{1}{k}$.

Now let us prove the sufficiency implication. Assume that $\lambda = (t, x, y) \in T$ appears in $T^\star$ without having been swapped with itself. Then, by construction of ReachLocations, $\lambda \in T^\star$ must have been formed as the result of swapping a triple $(t, x', y') \in T$ with a triple (t', x, y) from another original trajectory, where $(x', y') \neq (x, y)$. Buth then T would contain two triples with the same timestamp t and different spatial locations, which is a contradiction. □

Theorem 2 *The ReachLocations method achieves location k-diversity.*

Proof: Assume the adversary knows a sub-trajectory S of an original trajectory T. The sequence of timestamps in S allows the adversary to re-identify the anonymized trajectory $T^\star$ corresponding to T (because the timestamp sequence is preserved). By Proposition 2, any triple $\lambda \in T^\star \setminus S$ belongs to $T \setminus S$ with probability at most $\frac{1}{k}$. Now, consider a triple $\lambda = (t, x, y) \in T^{\star\star} \setminus S$, where $T^{\star\star}$ is an anonymized trajectory different from $T^\star$. The probability that λ came to $T^{\star\star} \setminus S$ from $T \setminus S$ is the probability that λ was swapped and swapping did not alter it. This probability is zero, because swaps preserve time coordinates but take place only between triples having different space coordinates. Hence, in terms of Definition 6, $Pr_\lambda[T|S] \leq \frac{1}{k}$ for every triple (T, S, λ) such that $T \in \mathcal{T}$, $S \preceq T$ and $\lambda \notin S$. □

Note that the previous proof also implies that, even if a triple $\lambda = (t, x, y) \notin S$ is shared by $M > 1$ anonymized trajectories, the probability of $\lambda \in T \setminus S$ remains at most $\frac{1}{k}$. What can be inferred by the adversary, however, is that M original trajectories (in general not the ones corresponding to the M anonymized trajectories) visited spatial coordinates (x, y) at possibly different times. Indeed, (t, x, y) can be obtained by swapping (t', x, y) and (t, x', y') for any t' such that $|t' - t| \leq R^t$ and for any $(x', y') \neq (x, y)$ at path distance at most R^s. If M is the total number of anonymized trajectories, then the adversary can be sure that original trajectory T visited spatial coordinates (x, y) at some time t' such that $|t' - t| \leq R^t$. Such inference by the adversary does not violate location k-diversity: violation would require guessing *both* the spatial *and* temporal coordinates of a triple in $T \setminus S$. Of course, the time threshold R^t must be taken large enough so that the time coordinate t is sufficiently protected.

7 Experimental results

We implemented SwapLocations and ReachLocations. SwapLocations performs clustering of trajectories using the partitioning step of the MDAV microaggregation heuristic [13]. We used two data sets in our experiments:

- *Synthetic data set.* We used Brinkhoff's generator [8] to generate 1,000 synthetic trajectories which altogether visit 45,505 locations in the German city of Oldenburg. Synthetic trajectories generated with Brinkhoff's generator have also been used in [1,33,34,47]. We used this data set mainly for comparing our methods with (k, δ)-anonymity [1]. The number of trajectories being moderate, we were able to run in reasonable time the methods to be compared with a large number of different parameter choices. Another advantage is that the street graph of Oldenburg was available, which is necessary to run ReachLocations. The downside of this data set having a moderate number of trajectories is that these are rather sparse, which causes the relative distortion in the anonymized data set to be substantial, no matter the method used. Anyway, this is not a serious problem to compare methods with each other.
- *Real-life data set.* We also used a real-life data set of cab mobility traces that were collected in the city of San Francisco [38]. This data set consists of 536 files, each of them containing the GPS coordinates of a cab during a period of time. After a filtering process, we obtained 4582 trajectories and 94 locations per trajectory on average. The advantage of this data set over the synthetic one is that it contains a larger number of trajectories and that these are real ones. Then, we show through a real example how appropriate is our distance metric for trajectory clustering. Also, we present utility measures on the SwapLocations method for this real-life data set using different space thresholds. The weakness of this data set is that it cannot be used for ReachLocations, because it does not include the underlying street graph of San Francisco.

7.1 Results on synthetic data

For the sake of reproducibility, we indicate the parameters we used in Brinkhoff's generator to generate our Oldenburg synthetic data set: 6 moving object classes and 3 external object classes; 10 moving objects and 1 external object generated per timestamp; 100 timestamps; speed 250; and "probability" 1,000. This resulted in 1,000 trajectories containing 45,405 locations. The maximum trajectory length was 100 points, the average length was 45.4 locations, and the median length was 44 locations.

7.1.1 Implementation details of our methods

We have introduced a new distance measure between trajectories used by the SwapLocations proposal during the clustering process. As mentioned in Section 5.1 above, our distance function can only be used within one of the connected components of the distance graph G. During the construction of the distance graph for the synthetic data we found 11 connected components, 10 of them of size 1. Therefore, we removed these 10 trajectories in order to obtain a new distance graph with just one connected component. In this way, we preserved 99% percent of all trajectories before the anonymization process. The removed trajectories were in fact trajectories of length one, *i.e.*, with just one location in each one.

The SwapLocations method has been implemented using the following simple microaggregation method for trajectories: first, create clusters of size k with minimum intra-cluster distance and then disperse the up to $k-1$ unclustered trajectories to existing clusters while minimizing the intra-cluster distance. This algorithm incurs no additional discarding of trajectories.

On the other hand, the ReachLocations method does not remove trajectories, unlike the SwapLocations method. It does, however, remove non-swappable locations, which causes the removal of any trajectory consisting of non-swappable locations only.

7.1.2 Implementing (k, δ)-anonymity for comparison with our method

We compared our proposals with (k, δ)-anonymity [1]. Since (k, δ)-anonymity only works over trajectories having the same time span, first a pre-processing step to partition the trajectories is needed. Superimposing the begin and end times of the trajectories through reduction of the time coordinate modulo a parameter π does not always yield at least k trajectories having the same time span; it may also happen that a trajectory disappears because the new reduced end time lies before the new reduced begin time.

We have used $\pi = 3$ which kept the maximum (and so discarded the minimum) trajectories. From the 1,000 synthetic trajectories, 40 were discarded because the end time was less than the begin time and 187 were discarded because there were at most 4 trajectories having the same time span. In total, 227 (22.7%) trajectories were discarded just in the pre-processing step. The remaining 773 trajectories were in 32 sets having the same time span, each set containing a minimum of 15 trajectories and 24 on average.

We performed (k, δ)-anonymization for $k = 2$, 4, 6, 8, 10, and 15 and $\delta = 0$, 1000, 2000, 3000, 4000 and 5000. Because of the pre-processing step, using a higher k was impossible without causing a significant number of additional trajectories to be discarded.

7.1.3 Utility comparison

The performance of our proposals strongly depends on the values of the time and space threshold parameters, denoted as R^t and R^s, respectively. In practice, these values must be chosen to maximize utility while affording sufficient privacy protection. Too large thresholds reduce utility (large space distortion if R^s is too high and large time distortion is R^t is too high), but too small thresholds reduce utility because of removal of many unswappable locations. As a rule of thumb, as illustrated in Example 1, the space threshold R^s must be sufficiently large so that within a radius R^s of any spatial location there are sufficiently distinct locations (*e.g.* if (x, y) lies in Tahrir Square, Cairo, there should be points outside the Square within a radius R^s of (x, y)).

In order to compute the total space distortion, a value for Ω must be chosen and this can be a challenging task. Note that the value of Ω is application-dependent, *e.g.* for applications where the distortion should measure the accuracy of trajectories Ω should be zero (only non-removed triples contribute to $TotalSD$), while for applications that should avoid removing any triples, Ω should be very high. That is why we propose to compare separately the following three utility properties: i) total space distortion; ii) percentage of removed trajectories; and iii) percentage of removed locations. To do so, we set $\Omega = 0$ when computing the total space distortion. Consequently, the percentage of removed triples as well as the percentage of removed trajectories are considered separately from the total space distortion.

It should be remarked that the computation of the total space distortion of the ReachLocations method is done using the Euclidean distance between locations rather than the distance defined by the reachability constraints (distance on the underlying network). Note that reachability constraints should be considered during the anonymization process but not necessarily when computing the total space distortion.

For successive anonymizations aimed at comparing the SwapLocations and ReachLocations methods with (k, δ)-anonymity, we set R^t and R^s in a way to obtain roughly the same total space distortion values as in (k, δ)-anonymity (see Table 1) with $\Omega = 0$. The idea is that, after making sure the three methods achieve roughly the same total space distortion, we will be able to focus on other utility properties like the percentage of removed trajectories and the percentage of removed locations. It should be remarked that our comparison is not entirely fair for any of the three methods because all of them are aimed at achieving different privacy notions. However, we believe that our results are indicative of the weaknesses and the strengths of our proposals.

Table 1
Total space distortion (TotalSD) of (k, δ)-anonymity for several parameter values (e6 stands for $\times 10^6$)

$\delta \setminus k$	2	4	6	8	10	15
0	48e6	93e6	120e6	143e6	165e6	199e6
1,000	19e6	60e6	86e6	109e6	131e6	165e6
2,000	4e6	32e6	56e6	78e6	99e6	133e6
3,000	.9e6	14e6	32e6	52e6	71e6	104e6
4,000	.2e6	5e6	16e6	32e6	48e6	79e6
5,000	.03e6	2e6	7e6	18e6	31e6	58e6

The above principle of equating the space distortions with (k, δ)-anonymity yields a value for the space threshold R^s in each of SwapLocations and ReachLocations; however, it does not constrain the time threshold, which we set at $R^t = 100$. Regarding R^s, we set it to achieve the total space distortions of (k, δ)-anonymity for cluster size $k = \{2, 4, 6, 8, 10, 15\}$ and

$$\delta = \{0, 1000, 2000, 3000, 4000, 5000\}$$

(parameter values considered in Table 1). In order to find such space thresholds efficiently, we assume that the total space distortions of our methods define a monotonically increasing function of the space threshold, *i.e.* the higher the space threshold, the higher the total space distortion. Under this assumption, we perform a logarithmic search over the set of space thresholds defined by the interval $[0, 10^6]$. The reason behind defining the maximum value for the space threshold as 10^6 is that it is high enough to achieve low numbers of removed trajectories. Indeed, as shown in Figure 2, for both methods there exists a value $R^s_{cutoff} < 10^6$ such that, for every space threshold $R^s > R^s_{cutoff}$, neither the total space distortion nor the percentage of removed locations and removed trajectories significantly change. Table 2 and Table 3 show the values of space thresholds used in each configuration of (k, δ)-anonymity for SwapLocations and ReachLocations, respectively.

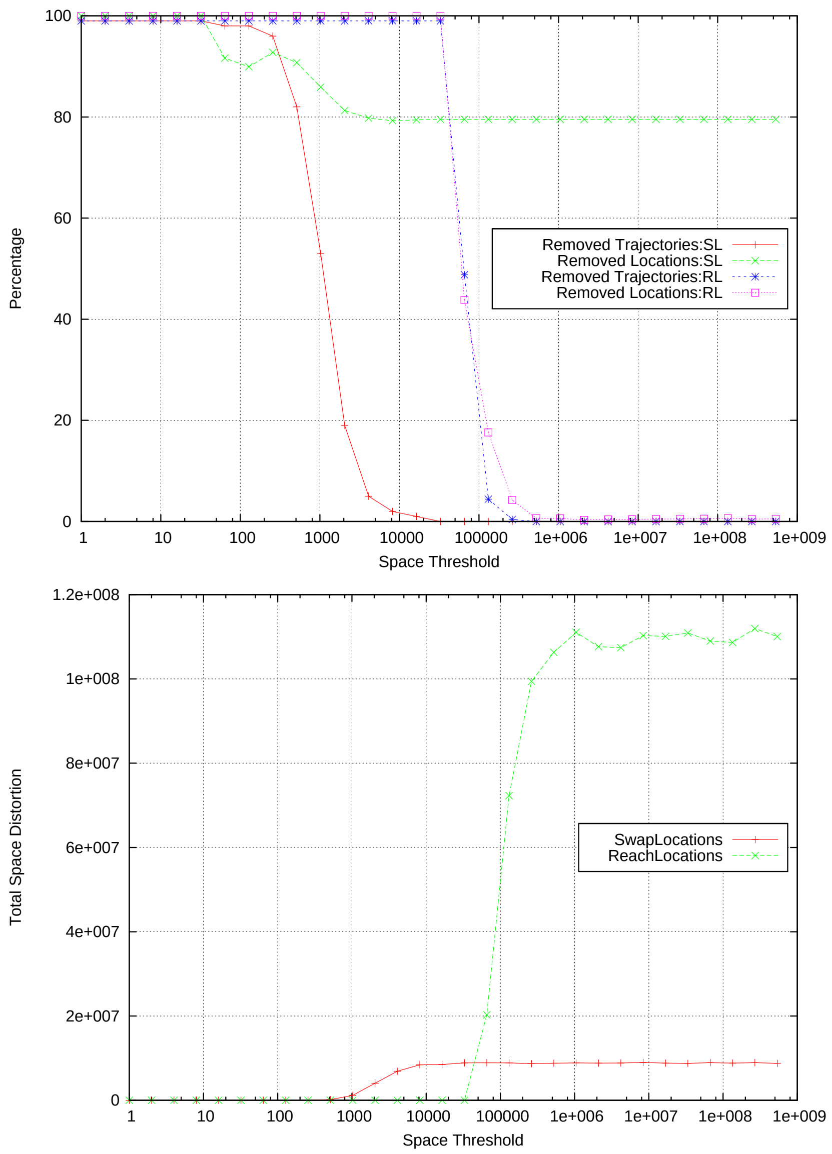


Fig. 2. Top, percentage of removed trajectories and locations with $k = 10$, $R^t = 100$ and several values of R^s for SwapLocations (SL) and ReachLocations (RL). Bottom, total space distortion with $k = 10$, $R^t = 100$ and several values of R^s for SwapLocations and ReachLocations

As it can be seen in Tables 2 and 3, we use the maximum value (10^6) of the space threshold for several configurations. This is because in those configurations the total space distortion caused by the (k, δ)-anonymity could not be reached by our methods no matter how much we increased the space threshold. Figure 3 explains this behavior by showing the values of total space distortion SwapLocations and ReachLocations minus the total space distortion of (k, δ)-anonymity. With almost every configuration, our methods have a total space distortion lower than the total space distortion of (k, δ)-anonymity. In the case of SwapLocations, the total space distortion is even much lower.

Table 2
Space thresholds used in SwapLocations to match the total space distortion of each configuration of (k, δ)-anonymity

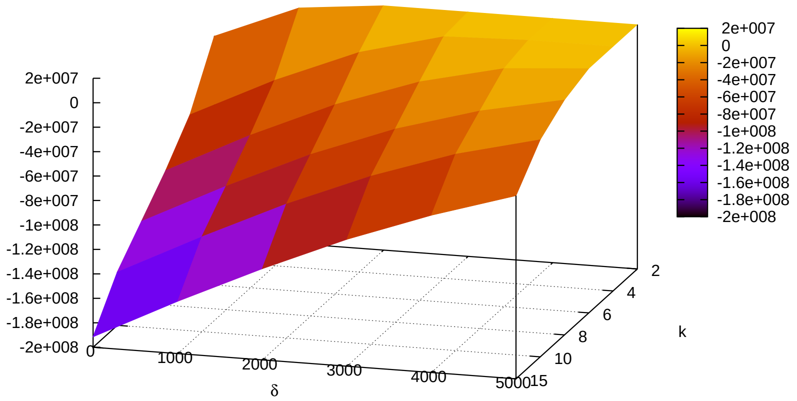
$\delta \setminus k$	2	4	6	8	10	15
0	10^6	10^6	10^6	10^6	10^6	10^6
1,000	10^6	10^6	10^6	10^6	10^6	10^6
2,000	899	10^6	10^6	10^6	10^6	10^6
3,000	257	10^6	10^6	10^6	10^6	10^6
4,000	86	1390	10^6	10^6	10^6	10^6
5,000	19	681	2507	10^6	10^6	10^6

Table 3
Space thresholds used in ReachLocations to match the total space distortion of each configuration of (k, δ)-anonymity

$\delta \setminus k$	2	4	6	8	10	15
0	499875	10^6	10^6	10^6	10^6	10^6
1,000	25090	106126	270157	10^6	10^6	10^6
2,000	4780	52468	93717	151915	249999	10^6
3,000	749	37124	64801	95585	132857	238884
4,000	136	25540	51089	73088	94465	152862
5,000	57	18059	39061	58584	79101	113280

In general, SwapLocations does not reach high values of the total space distortion because it removes more locations than ReachLocations in order to achieve trajectory k-anonymity. Note that removing locations does not increase the total space distortion because we are considering $\Omega = 0$. Tables 4 and 5 show in detail the percentage of removed trajectories and the percentage of removed locations for different values of $k = \{2, 4, 6, 8, 10, 15\}$ and $\delta = \{0, 1000, 2000, 3000, 4000, 5000\}$, for SwapLocations and ReachLocations, respectively.

As it can be seen in Table 4, in general SwapLocations removes less trajectories than (k, δ)-anonymity because SwapLocations can cluster non-overlapping trajectories. Indeed, with (k, δ)-anonymity 227 trajectories were discarded in the pre-processing step alone because their time span could not match the time span of other trajectories, and additional outlier trajectories were discarded during clustering, up to a total 24% of discarded trajectories. However, SwapLocations removed up to 84% of all locations in the worst cases and thus, it may not be suitable for applications where preserving the number of locations really matters. SwapLocations removes any location whose swapping set



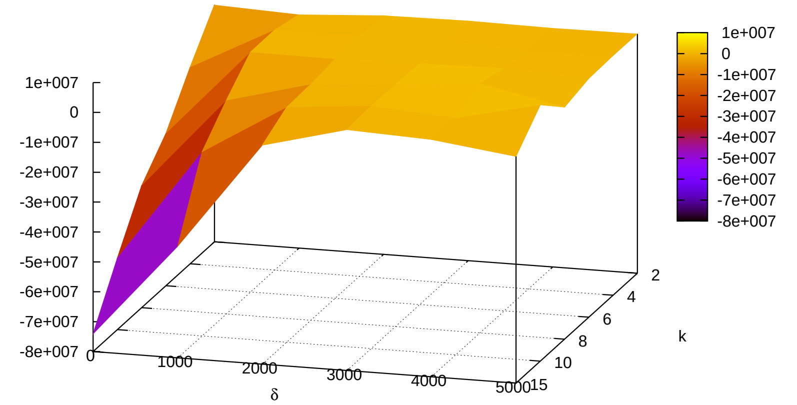


Fig. 3. Top: total space distortion of SwapLocations minus total space distortion of (k, δ)-anonymity for several parameter configurations. Bottom: total space distortion of ReachLocations minus total space distortion of (k, δ)-anonymity for several parameter configurations. The space thresholds defined in Tables 2 and 3 have been used, respectively.

.

U contains less than k locations, which is a relatively frequent event when k trajectories with different lengths are clustered together. As the cluster size k increases, the length diversity tends to increase and the removal percentage increases. A simple way around the location removal problem is to create clusters that contain trajectories with roughly the same length, even though this may result in a higher total space distortion; higher space distortion is a natural consequence of clustering based on the trajectory length rather than the trajectory distance.

Table 4
Percentage of trajectories (columns labeled with **T**) and locations (columns labeled **L**) removed by SwapLocations when using time threshold 100, $k = \{2, 4, 6, 8, 10, 15\}$ and space thresholds that match the space distortion caused by (k, δ)-anonymity with the previous k's and $\delta = \{0, 1000, 2000, 3000, 4000, 5000\}$. Percentages have been rounded to integers for compactness.

$\delta \setminus k$	2		4		6		8		10		15	
	T	**L**	**T**	**L**	**T**	**L**	**T**	**L**	**T**	**L**	**T**	**L**
0	0	34	0	58	0	69	1	75	0	79	0	84
1000	0	34	0	58	0	69	1	75	0	79	0	84
2000	4	45	0	58	0	69	1	75	0	79	0	84
3000	11	62	0	58	0	69	1	75	0	79	0	84
4000	19	68	5	66	0	69	1	75	0	79	0	84
5000	32	78	20	73	4	72	1	75	0	79	0	84

Table 5 shows that ReachLocations removes few trajectories when δ is small and k is large. The reason is that, for those parameterizations, (k, δ)-anonymity introduces so much total space distortion that ReachLocations can afford taking the maximum space threshold $R^s = 10^6$ without reaching that much distortion. Such a high space threshold allows ReachLocations to easily swap spatial coordinates, so that very few locations need to be removed. Furthermore, the trajectories output by ReachLocations are consistent with the underlying city topology. As said above, the only drawback of this method is that in general it does not provide trajectory k-anonymity; rather, it provides location k-diversity.

7.1.4 Spatio-temporal range queries

As stated in Section 3.2, a typical use of trajectory data is to perform spatio-temporal range queries on them. That is why we report empirical results when performing the two query types described and motivated in Section 3.2: *Sometime_Definitely_Inside* (SI) and *Always_Definitely_Inside* (AI). We accumulate the number of trajectories in a set of trajectories $\mathcal{T}$ that satisfy the SI or AI range queries using the SQL style code below.

- Query $\mathcal{Q}_1(\mathcal{T}, R, t_b, t_e)$:
  SELECT COUNT (*) FROM $\mathcal{T}$ WHERE SI($\mathcal{T}$.traj, R, $\mathtt{t_b}, \mathtt{t_e}$)
- Query $\mathcal{Q}_2(\mathcal{T}, R, t_b, t_e)$:
  SELECT COUNT (*) FROM $\mathcal{T}$ WHERE AI($\mathcal{T}$.traj, R, $\mathtt{t_b}, \mathtt{t_e}$)

Then, we define two different *range query distortions*:

Table 5
Percentage of trajectories (columns labeled with **T**) and locations (columns labeled **L**) removed by ReachLocations when using time threshold 100, $k = \{2, 4, 6, 8, 10, 15\}$ and space thresholds that match the space distortion caused by (k, δ)-anonymity with the previous k's and $\delta = \{0, 1000, 2000, 3000, 4000, 5000\}$. Percentages have been rounded to integers for compactness.

$\delta \setminus k$	2		4		6		8		10		15	
	T	**L**	**T**	**L**	**T**	**L**	**T**	**L**	**T**	**L**	**T**	**L**
0	0	1	0	3	0	3	0	4	0	4	0	3
1000	0	2	0	3	0	3	0	4	0	5	0	3
2000	36	27	9	18	3	11	0	5	0	6	0	4
3000	74	38	33	39	18	28	6	21	2	13	0	7
4000	82	43	65	49	41	40	20	34	10	27	2	16
5000	84	60	84	53	60	52	40	44	27	35	10	27

- $\mathrm{SID}(\mathcal{T}, \mathcal{T}^\star) = \frac{1}{|\xi|} \sum_{\forall < R, t_b, t_e > \in \xi} \frac{|\mathcal{Q}_1(\mathcal{T}, R, t_b, t_e) - \mathcal{Q}_1(\mathcal{T}^\star, R, t_b, t_e)|}{\max(\mathcal{Q}_1(\mathcal{T}, R, t_b, t_e), \mathcal{Q}_1(\mathcal{T}^\star, R, t_b, t_e))}$ where ξ is a set of SI queries as defined in Section 3.2 (definition of SI adapted to non-uncertain trajectories).
- $\mathrm{AID}(\mathcal{T}, \mathcal{T}^\star) = \frac{1}{|\xi|} \sum_{\forall < R, t_b, t_e > \in \xi} \frac{|\mathcal{Q}_2(\mathcal{T}, R, t_b, t_e) - \mathcal{Q}_2(\mathcal{T}^\star, R, t_b, t_e)|}{\max(\mathcal{Q}_2(\mathcal{T}, R, t_b, t_e), \mathcal{Q}_2(\mathcal{T}^\star, R, t_b, t_e))}$ where ξ is a set of AI queries as defined in Section 3.2 (definition of AI adapted to non-uncertain trajectories).

For our experiments with the synthetic data set, we chose random time intervals $[t_b, t_e]$ such that $0 \leq t_e - t_b \leq 10$. Also, we chose random uncertain trajectories with a randomly chosen radius $0 \leq \sigma \leq 750$ as regions R. Actually, 10 and 750 are, respectively, roughly a quarter of the average duration and distance of all trajectories. Note that we used uncertain trajectories *only* as regions R; however, the methods we are considering in this paper all release non-uncertain trajectories.

Armed with these settings, we ran $100,000$ times both queries $\mathcal{Q}_1$ and $\mathcal{Q}_2$ on the original data set and the anonymized data sets provided by SwapLocations, ReachLocations, and (k, δ)-anonymity; that is, we took a set ξ with $|\xi| = 100,000$. The ideal range query distortion would be zero, which means that query $\mathcal{Q}_i$ for $i \in 1, 2$ yields the same result for both the original and the anonymized data sets; in practice, zero distortion is hard to obtain. Therefore, in order to compare our methods against (k, δ)-anonymity, we use the same parameters of the previous experiments (Tables 1, 2, and 3). We show in Tables 6 and 7 a comparison of SwapLocations, respectively ReachLocations, against (k, δ)-anonymity in terms of SID and AID.

Table 6

Range query distortion of SwapLocations compared to (k,δ)-anonymity for SID (columns labeled with **S**) and AID (columns labeled with **A**) when using $k = \{2,4,6,8,10,15\}$ and space thresholds that match the space distortion caused by (k,δ)-anonymity with the previous k's and $\delta = \{0,1000,2000,3000,4000,5000\}$. In this table, a range query distortion x obtained with SwapLocations and a range query distortion y obtained with (k,δ)-anonymity are represented as the integer rounding of $(y-x)*100$. Hence, values in the table are positive if and only if SwapLocations outperforms (k,δ)-anonymity.

$\delta \setminus k$	2		4		6		8		10		15	
	S	**A**	**S**	**A**	**S**	**A**	**S**	**A**	**S**	**A**	**S**	**A**
0	34	29	31	14	36	16	36	13	37	13	43	14
1000	24	20	24	8	28	10	27	8	28	9	41	14
2000	18	14	18	4	20	3	20	2	27	6	39	10
3000	8	3	11	−2	13	0	16	−1	21	4	36	10
4000	−6	−7	6	−6	9	−5	11	−4	17	2	30	5
5000	−22	−19	1	−9	3	−9	7	−7	14	−2	27	2

Table 7

Range query distortion of ReachLocations compared to (k,δ)-anonymity for SID (columns labeled with **S**) and AID (columns labeled with **A**) when using $k = \{2,4,6,8,10,15\}$ and space thresholds that match the space distortion caused by (k,δ)-anonymity with the previous k's and $\delta = \{0,1000,2000,3000,4000,5000\}$. In this table, a range query distortion x obtained with ReachLocations and a range query distortion y obtained with (k,δ)-anonymity are represented as the integer rounding of $(y-x)*100$. Hence, values in the table are positive if and only if ReachLocations outperforms (k,δ)-anonymity.

$\delta \setminus k$	2		4		6		8		10		15	
	S	**A**	**S**	**A**	**S**	**A**	**S**	**A**	**S**	**A**	**S**	**A**
0	34	25	28	12	33	10	32	5	31	5	37	6
1000	25	19	21	6	24	4	23	1	25	2	35	5
2000	10	10	8	−7	17	−3	19	−3	23	−3	33	4
3000	−4	2	0	−12	9	−12	13	−5	19	−4	29	1
4000	−11	−6	−6	−18	−2	−17	3	−16	13	−6	26	−3
5000	−14	−5	−10	−22	−8	−25	−4	−21	8	−14	20	−5

It can be seen from Table 6 that SwapLocations performs significantly better than (k, δ)-anonymity for every cluster size and $\delta \leq 3000$. On the other hand, Table 7 shows that ReachLocations outperforms (k, δ)-anonymity only for δ up to roughly 2000. Not surprisingly, SwapLocations offers better performance than ReachLocations, because the latter must deal with reachability constraints. It is also remarkable that ReachLocations performs much better in terms of SID than in terms of AID. The explanation is that, while (k, δ)-anonymity and SwapLocations operate at the trajectory level, ReachLocations works at the location level.

We conclude that, according to these experiments, our methods perform better than (k, δ)-anonymity regarding range query distortion for values of δ up to 2000. The performance for larger values of δ is less and less relevant: indeed, when $\delta \to \infty$, (k, δ)-anonymity means that no trajectory needs to be anonymized and hence the anonymized trajectories are the same as the original ones.

7.2 Results on real-life data

The San Francisco cab data set [38] we used consists of several files each of them containing the GPS information of a specific cab during May 2008. Each line within a file contains the space coordinates (latitude and longitude) of the cab at a given time. However, the mobility trace of a cab during an entire month can hardly be considered a single trajectory. We used big time gaps between two consecutive locations in a cab mobility trace to split that trace into several trajectories. All trajectory visualizations shown in this Section were obtained using Google Earth.

For our experiments we considered just one day of the entire month given in the real-life data set, but the empirical methodology described below could be extended to several days. In particular, we chose the day between May 25 at 12:04 hours and May 26 at 12:04 hours because during this 24-hour period there was the highest concentration of locations in the data set. We also defined the maximum time gap in a trajectory as 3 minutes; above 3 minutes, we assumed that the current trajectory ended and that the next location belonged to a different trajectory. This choice was based on the average time gap between consecutive locations in the data set, which was 88 seconds; hence, 3 minutes was roughly twice the average. In this way, we obtained 4582 trajectories and 94 locations per trajectory on average.

The next step was to filter out trajectories with strange features (outliers). These outliers could be detected based on several aspects like velocity, city topology, etc. We focused on velocity and defined 240 km/h as the maximum speed that could be reached by a cab. Consequently, the distance between two consecutive locations could not be greater than 12 km because the maximum within-trajectory time gap was 3 minutes. This allowed us to detect and remove trajectories containing obviously erroneous locations; Figure 4 shows one of these removed outliers where a cab appeared to have jumped far into the sea probably due to some error in recording its GPS coordinates. Altogether, we removed 45 outlier trajectories and we were left with a data set of 4547 trajectories with an average of 93 locations per trajectory. Figure 5 shows the ten longest trajectories (in number of locations) in the final data set that we used.

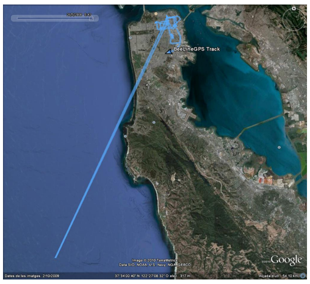


Fig. 4. Example of an outlier trajectory in the original real-life data set

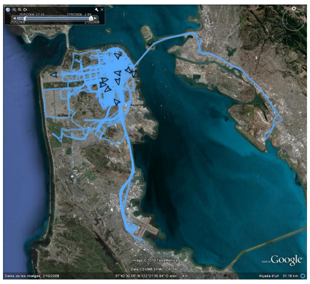


Fig. 5. Ten longest trajectories in the filtered real-life data set

7.2.1 *Experiments with the distance metric*

We propose in this paper a new distance metric designed specifically for clustering trajectories. Our distance metric considers both space and time, dealing even with non-overlapping or partially-overlapping trajectories. Contrary to the synthetic data where 10 trajectories had to be removed because the distances to them could not be computed, in this real-life data set our distance function could be computed for every pair of trajectories.

Figure 6 shows two trajectories identified by our distance metric as the two closest ones in the data set. The two cabs moved around a parking lot and therefore stayed very close to one another in space. Also in time both trajectories were very close: one of them was recorded between 12:00:49 hours and 13:50:47 hours, while the other was recorded between 12:00:25 hours and 13:52:30 hours. Therefore, both trajectories were correctly identified by our distance metric as being close in time and space; they could be clustered together with minimum utility loss for anonymization purposes.

To compare, Figure 7 shows two trajectories identified by the Euclidean distance as the two closest ones in the data set. These trajectories are located in a parking lot inside San Francisco Airport and, spatially, they are closer

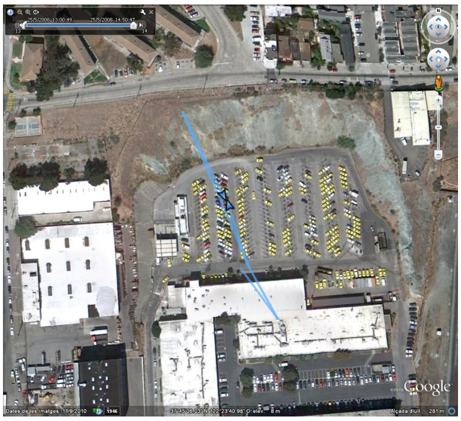


Fig. 6. The two closest trajectories in the real-life data set according to our distance metric

than the two trajectories shown in Figure 6. However, one of these trajectories was recorded between 24:42:55 hours and 24:55:59 hours, while the other was recorded between 19:05:29 hours and 19:06:15 hours. Hence, they should not be in the same cluster, because an adversary with time knowledge can easily distinguish them.

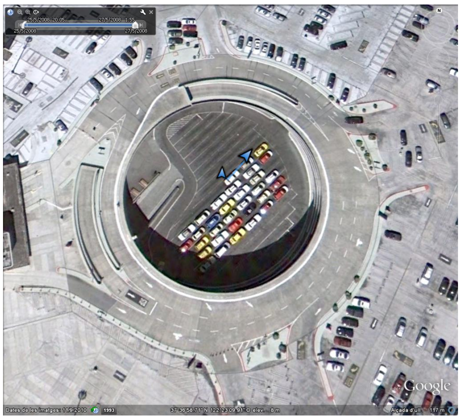


Fig. 7. The two closest trajectories in the real-life data set according to the Euclidean distance

7.2.2 Experiments with the SwapLocations method

The ReachLocations method cannot be used when the graph of the city is not provided. Hence, in the experiments with the San Francisco real data we just considered the SwapLocations method. As in the experiments with synthetic data, we set $\Omega = 0$ during the computation of the total space distortion. Figure 8 shows the values of total space distortion given by the SwapLocations for different space thresholds and different cluster sizes.

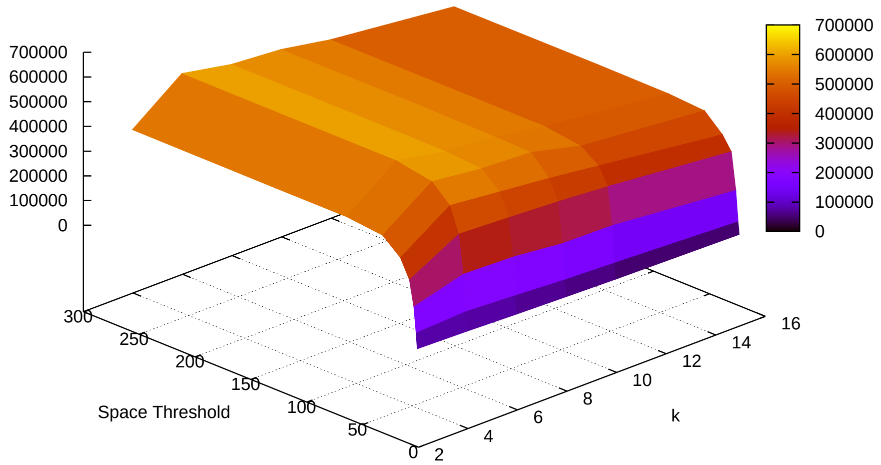


Fig. 8. Total space distortion (km) for SwapLocations using several different space thresholds and cluster sizes on the real-life data set

Two other utility properties we are considering in this work are: percentage of removed trajectories and percentage of removed locations. Table 8 shows the values obtained with the SwapLocations method for both utility properties.

Finally, Table 9 reports the performance of SwapLocations regarding spatio-temporal range queries. We picked random time intervals of length at most 20 minutes. Also, random uncertain trajectories with uncertainty threshold of size at most 7 km were chosen as the regions. Analogously to the experiments with the synthetic data set, 20 and 7 are roughly a quarter of the average duration and distance of all trajectories, respectively. It can be seen that the SwapLocations method provides low range query distortion for every value of k when the space threshold is small, *i.e.* when the total space distortion is also small. However, the smaller the space threshold, the larger the number of removed trajectories and locations (see Table 8). This illustrates the trade-off between the utility properties considered.

Table 8

Percentage of trajectories (columns labeled with **T**) and locations (columns labeled with **L**) removed by SwapLocations for several values of k and several space thresholds R^s on the real-life data set. Percentages have been rounded to integers for compactness.

$R^s \setminus k$	2		4		6		8		10		15	
	T	**L**	**T**	**L**	**T**	**L**	**T**	**L**	**T**	**L**	**T**	**L**
1	23	43	40	64	49	71	58	74	62	77	71	81
2	19	29	34	47	42	54	50	58	54	60	50	66
4	14	17	27	29	35	35	40	40	45	41	54	49
8	9	10	19	19	25	25	31	29	34	31	42	38
16	5	7	11	16	17	22	20	27	23	30	32	38
32	1	7	2	15	3	22	4	27	5	30	8	38
64	0	6	0	15	0	22	0	27	0	30	0	38
128	0	6	0	15	0	22	0	27	0	30	0	38

Table 9

Range query distortion caused by SwapLocations on the real-life data set for SID (columns labeled with **S**) and AID (columns labeled with **A**), for several values of k and several space thresholds R^s. In this table, a range query distortion x is represented as the integer rounding of $x * 100$ for compactness.

$R^s \setminus k$	2		4		6		8		10		15	
	S	**A**	**S**	**A**	**S**	**A**	**S**	**A**	**S**	**A**	**S**	**A**
1	13	22	18	27	20	29	19	29	24	31	25	34
2	16	24	25	34	26	35	24	35	27	37	27	37
4	18	25	30	37	33	41	34	42	38	46	38	45
8	21	27	34	40	38	44	40	46	44	50	48	54
16	20	26	36	42	42	47	45	50	50	54	53	58
32	21	26	39	44	45	49	48	53	53	57	58	62
64	20	25	39	44	46	50	51	54	54	57	61	64
128	21	26	39	44	48	50	51	56	54	58	61	64

8 Conclusions and future work

We have presented two permutation-based heuristic methods to anonymize trajectories with the common features that: i) places and times in the anonymized trajectories are true original places and times with full accuracy; ii) both methods can deal with trajectories with partial or no time overlap, thanks to a new distance also introduced in this paper. The first method aims at trajectory k-anonymity while the second method takes reachability constraints into account, that is, it assumes a territory constrained by a network of streets or roads; to avoid removing too many locations, the second method changes its privacy ambitions from trajectory k-anonymity to location k-diversity.

Both methods use permutation of locations, and the first method uses also trajectory microaggregation. There are few counterparts in the literature comparable to the first method, and virtually none comparable to the second method. Experimental results show that, for most parameter choices and for similar privacy levels, our methods offer better utility than (k, δ)-anonymity.

Future work will be directed towards designing trajectory anonymization methods aimed at achieving trajectory p-privacy (see Definition 6), but discarding less locations than the SwapLocations method. Also, finding trajectory anonymization methods for constrained territories with better utility than ReachLocations is an open challenge.

Acknowledgments

Thanks go to Michal Sramka for helping with the initial state of the art of this work, while he was with us. We acknowledge useful comments by three anonymous reviewers, which led to substantial improvement of the initial versions of this paper. In particular, we are grateful to one of the reviewers for motivating Example 1.

References

[1] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: uncertainty for anonymity in moving objects databases. In *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008*, Cancun, Mexico, 7-12 April 2008, pages 376–385. IEEE, 2008.

[2] O. Abul, F. Bonchi, and M. Nanni. Anonymization of moving objects databases by clustering and perturbation. *Information Systems*, 35(8):884–910, 2010.

[3] C. C. Aggarwal and P. S. Yu. A condensation approach to privacy preserving data mining. In *Proceedings of the 9th International Conference on Extending Database Technology, EDBT 2004*, Heraklion, Crete, Greece, 14-18 March 2004, volume 2992 of *Lecture Notes in Computer Science*, pages 183–199. Springer, 2004.

[4] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(1-2):75–91, 1995.

[5] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):209–216, 1991.

[6] F. Bonchi. Privacy preserving publication of moving object data. In *Privacy in Location-Based Applications, Research Issues and Emerging Trends*, volume 5599 of *Lecture Notes in Computer Science*, pages 190–215. Springer, 2009.

[7] F. Bonchi, Y. Saygin, V. S. Verykios, M. Atzori, A. Gkoulalas-Divanis, S. V. Kaya, and E. Savas. Privacy in spatiotemporal data mining. In *Mobility, Data Mining and Privacy*, pages 297–333. Springer, 2008.

[8] T. Brinkhoff. Generating traffic data. *IEEE Data Engineering Bulletin*, 26(2):19–25, 2003.

[9] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of 2005 ACM SIGMOD International Conference on Management of Data*, Baltimore, Maryland, USA, 14-16 June 2005, pages 491–502. ACM, 2005.

[10] J. Domingo-Ferrer and U. González-Nicolás. Hybrid microdata using microaggregation. *Information Sciences*, 180(15):2834–2844, 2010.

[11] J. Domingo-Ferrer and J. M. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering*, 14(1):189–201, 2002.

[12] J. Domingo-Ferrer, M. Sramka, and R. Trujillo-Rasúa. Privacy-preserving publication of trajectories using microaggregation. In *Proceedings of the SIGSPATIAL ACM GIS 2010 International Workshop on Security and Privacy in GIS and LBS, SPRINGL 2010*, San Jose, California, USA, 2 November 2010. ACM, 2010.

[13] J. Domingo-Ferrer and V. Torra. Ordinal, continuous and heterogenerous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11(2):195–212, 2005.

[14] J. Domingo-Ferrer, F. Sebé and A. Solanas. A polynomial-time approximation to optimal multivariate microaggregation. *Computers & Mathematics with Applications*, 55(4):714-732, 2008.

[15] J. Domingo-Ferrer and V. Torra. A critique of k-anonymity and some of its enhancements. In *Proceedings of the 3rd International Conference on Availability, Reliability and Security, ARES 2008*, Barcelona, Spain, 4-7 March 2008, pages 990–993. IEEE, 2008.

[16] R. W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345–350, 1962.

[17] L. Forlizzi, R. H. Güting, E. Nardelli, and M. Schneider. A data model and data structures for moving objects databases. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD 2000*, Dallas, Texas, USA, 16-18 May 2000, pages 319–330. ACM, 2000.

[18] B. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: a survey on recent developments. *ACM Computing Surveys*, 42(4), art. no. 14, 2010.

[19] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services, MobiSys 2003*, San Francisco, California, USA, 5-8 May 2003. USENIX, 2003.

[20] M. Gruteser and B. Hoh. On the anonymity of periodic location samples. In *Proceedings of the 2nd International Conference on Security in Pervasive Computing, SPC 2005*, Boppard, Germany, 6-8 April 2005, volume 3450 of *Lecture Notes in Computer Science*, pages 179–192. Springer, 2005.

[21] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *Proceedings of the IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communication Networks, SecureComm 2005*, Athens, Greece, 5-9 September 2005. IEEE, 2005.

[22] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007*, Alexandria, Virginia, USA, 28-31 October 2007, pages 161–171. ACM, 2007.

[23] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Achieving guaranteed anonymity in gps traces via uncertainty-aware path cloaking. *IEEE Transactions on Mobile Computing*, 9(8):1089–1107, 2010.

[24] H. Hu, J. Xu, S. T. On, J. Du, and J. Kee-Yin Ng. Privacy-aware location data publishing. *ACM Transactions on Database Systems*, 35(3), art. no. 18, 2010.

[25] H. Hu, J. Xu, and D. L. Lee. PAM: An efficient and privacy-aware monitoring framework for continuously moving objects. *IEEE Transactions on Knowledge and Data Engineering*, 22(3):404–419, 2010.

[26] E. Kaplan, T. B. Pedersen, E. Savas, and Y. Saygin. Privacy risks in trajectory data publishing: reconstructing private trajectories from continuous properties. In *Proceedings of the 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, KES 2008, Part II*,

Zagreb, Croatia, 3-5 September 2008, volume 5178 of *Lecture Notes in Computer Science*, pages 642–649. Springer, 2008.

[27] E. Kaplan, T. B. Pedersen, E. Savas, and Y. Saygin. Discovering private trajectories using background information. *Data and Knowledge Engineering*, 69(7):723–736, 2010.

[28] N. Li, T. Li, and S. Venkatasubramanian. t-Closeness: privacy beyond k-anonymity and ℓ-diversity. In *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007*, Istanbul, Turkey, 15-20 April 2007, pages 106–115. IEEE, 2007.

[29] T. W. Liao. Clustering of time series data - a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.

[30] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: privacy beyond k-anonymity. In *Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006*, Atlanta, Georgia, USA, 3-8 April 2006, pages 24–35. IEEE, 2006.

[31] N. Mohammed, B. C. M. Fung, and M. Debbabi. Walking in the crowd: anonymizing trajectory data for pattern analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009*, Hong Kong, China, 2-6 November 2-6 2009, pages 1441–1444. ACM, 2009.

[32] A. Monreale, G. Andrienko, N. Andrienko, F. Giannotti, D. Pedreschi, S. Rinzivillo, and S. Wrobel. Movement data anonymity through generalization. *Transactions on Data Privacy*, 3(2):91–121, 2010.

[33] M. E. Nergiz, M. Atzori, and Y. Saygin. Towards trajectory anonymization: a generalization-based approach. In *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS, SPRINGL 2008*, Irvine, California, USA, 4 November 2008, pages 52–61. ACM, 2008.

[34] M. E. Nergiz, M. Atzori, Y. Saygin, and B. Guc. Towards trajectory anonymization: a generalization-based approach. *Transactions on Data Privacy*, 2(1):47–75, 2009.

[35] M. E. Nergiz, C. Clifton, and A. E. Nergiz. Multirelational k-anonymity. In *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007*, Istanbul, Turkey, 15-20 April 2007, pages 1417–1421. IEEE, 2007.

[36] B. Palanisamy and L. Liu. MobiMix: Protecting location privacy with mix-zones over road networks. In *Proceedings of 27th International Conference on Data Engineering, ICDE 2011*, Hannover, Germany, 11-16 April 2011, pages 494–505. IEEE, 2011.

[37] R. G. Pensa, A. Monreale, F. Pinelli, and D. Pedreschi. Pattern-preserving k-anonymization of sequences and its application to mobility data mining. In *Proceedings of the 1st International Workshop on Privacy in Location-Based Applications, ESORICS-PiLBA 2008*, Malaga, Spain, 9 October 2008, volume 397 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[38] M. Piorkowski, N. Sarafijanovoc-Djukic, and M. Grossglauser. A parsimonious model of mobile partitioned networks with clustering. In *The First International Conference on COMmunication Systems and NETworkS (COMSNETS)*, Bangalore, India, January 2009.

[39] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, SRI Computer Science Laboratory, 1998.

[40] R. Shonkwiler. Computing the hausdorff set distance in linear time for any l(p) point distance. *Information Processing Letters*, 38(4):201–207, 1991.

[41] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[42] L. Sweeney. *Research Artifacts*, checked Jan. 27, 2012. `http://dataprivacylab.org/people/sweeney/artifacts.html`

[43] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *Proceedings of the 9th International Conference on Mobile Data Management, MDM 2008*, Beijing, China, 27-30 April 2008, pages 65–72. IEEE, 2008.

[44] G. Trajcevski, O. Ouri, K. Hinrichs, and S. Chamberlain. Managing uncertainty in moving objects databases. *ACM Trans. Database Syst.*, 29(3):463–507, 204.

[45] T. M. Truta and B. Vinay. Privacy protection: p-sensitive k-anonymity property. In *Proceedings of the 2nd International Workshop on Privacy Data Management, ICDE-PDM 2006*, Atlanta, Georgia, USA, 3-7 April 2006, pages 94–103. IEEE, 2006.

[46] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang. (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2006*, Philadelphia, Pennsylvania, USA, 20-23 August 2006, pages 754–759. ACM, 2006.

[47] R. Yarovoy, F. Bonchi, L. V. S. Lakshmanan, and W. H. Wang. Anonymizing moving objects: how to hide a mob in a crowd? In *Proceedings of the 12th International Conference on Extending Database Technology, EDBT 2009*, Saint Petersburg, Russia, 24-26 March 2009, volume 360 of *ACM International Conference Proceeding Series*, pages 72–83. ACM, 2009.

[48] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. Aggregate query answering on anonymized tables. In *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007*, Istanbul, Turkey, 15-20 April 2007, pages 116–125. IEEE, 2007.